TAMPERE UNIVERSITY OF TECHNOLOGY

EKI LEHTIMÄKI

NON-LINEAR STRUCTURAL ANALYSIS IN PERFORMANCE BASED FIRE - INTEGRATION OF DESIGN PROGRAMS

Master of Science Thesis

Examiners: Professor Markku Heinisuo,
Doctor Jyri Outinen
Subject and examiners approved by
the Built Environment Faculty Council
on 5.10.2011

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Civil Engineering
EKI LEHTIMÄKI : Non-linear structural analysis in performance based fire - integration of design programs
Master of Science Thesis, 73 pages
April 2012
Major: Structural Engineering
Examiners: professor Markku Heinisuo, Dr. Jyri Outinen (Ruukki Construction)
Keywords: Non-linear finite element analysis, Vulcan, natural fire design, fire safety engineering, steel structures, building information modelling, integrated design environment

The design of structures in fire has traditionally been based on ISO standard curves. For more accurate design, methods that simulate the real fires possible for the building in case should be preferred. Natural fire design concept is a collection of this kind of methods presented in literature and used in real construction projects.

Natural Fire Design environment (NFD-environment) is a ”software toolbox” aimed to integrate computer programs needed for performance-based structural fire safety engineering into one design environment easily accessible for structural engineer using natural fire design methods in real projects. NFD-environment has been developed in co-operation of Ruukki and Tampere University of Technology. Purpose of this master's thesis project was to extend non-linear structural finite element analysis capabilities of the environment by integrating structural analysis software Vulcan into it.

Analysing structure with linear finite element method and then reducing the resistance of members and joints according to the temperature reached in fire is a sufficient method, when structure is statically determined. For statically undetermined structures, non-linear finite element analysis is needed. Vulcan, being a special software to analyse structures in fire, is suitable for this kind of analysis. It could be integrated in the design environment by programming data transfer links with the building information modelling software and fire dynamics simulation software used in the NFD-environment.

Additional development was made also by creating algorithms to linearise and group temperature curves in order to speed up calculation time.

The key result of this thesis is a more advanced planning environment for structures in natural fire. Non-linear finite element analysis makes calculation of statically undetermined structures accurate, and temperature curve processing reduces the calculation time cost of non-linear analysis.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Rakennustekniikan koulutusohjelma
EKI LEHTIMÄKI: Epälineaarinen rakenneanalyysi toiminnallisessa palossa - suunnitteluohjelmien yhteensovittaminen
Diplomityö, 73 sivua
Huhtikuu 2012
Pääaine: Rakennesuunnittelu
Tarkastajat: professori Markku Heinisuo, TkT Jyri Outinen (Ruukki Construction)
Avainsanat: epälineaarinen elementtimenetelmä, Vulcan, luonnollinen palo, paloturvallisuustekniikka, teräsrakenteet, rakennusten tietomallit, integroitu suunnittelujärjestelmä

Rakenteiden palotekininen suunnittelu on perinteisesti pohjautunut ISO-standardipalokäyriin. Tarkempaan suunnitteluun päästäisiin käyttämällä menetelmiä, jotka simuloivat suunniteltavalle rakennukselle mahdollisia todellisia tulipaloja. Luonnolliseen paloon perustuva suunnittelukonsepti (Natural Fire Design Concept) on kokoelma kirjallisuudessa esitettyjä ja todellisissa rakennushankkeissa käytettyjä tällaisia menetelmiä.

Natural Fire Design (NFD) -ympäristö on kokoelma suunnitteluohjelmia. Sen tarkoituksena on sisällyttää kaikki toiminnalliseen palotekniseen rakennesuunnitteluun tarvittavat tietokone-ohjelmat yhteen suunnitteluympäristöön. NFD-ympäristö on kehitetty Ruukin ja Tampereen teknillisen yliopiston yhteistyönä. Tämän diplomityön tarkoituksena on parantaa ympäristön kykyä epälineaariseen elementtimenetelmään perustuvaan rakenneanalyysiin liittämällä Vulcan-ohjelma osaksi NFD-ympäristöä.

Rakenteen analysoiminen lineaarisella elementtimenetelmällä ja sauvojen ja liitosten kestävyyksien pienentäminen lopuksi tulipalossa saavutetun lämpötilan mukaiseksi riittää staattisesti määrätyille rakenteille. Staattisesti määräämättömän rakenteen tapauksessa tarkkaan ratkaisuun tarvitaan epälineaarista elementtimenetelmää. Vulcan, joka on erityisesti rakenneanalyysiin tulipalossa tarkoitettu ohjelma, sopii tällaiseen analyysiin. Se voitiin liittää osaksi NFD-suunnitteluympäristöä ohjelmoimalla tiedonsiirtolinkit NFD-ympäristöön kuuluvien rakennuksen tietomallinukseen käytettävän ohjelman ja palosimulointiohjelman kanssa.

Lisäksi kehitettiin algoritmit lämpötilakäyrien linearisoimiseen ja ryhmittelyyn laskenta-ajan pienentämiseksi.

Työn tärkein tulos on kehittyneempi suunnitteluympäristö rakenteiden toiminnalliseen palotekniseen suunnitteluun. Epälineaarisella elementtimenetelmällä staattisesti määräämättömien rakenteiden laskeminen on tarkempaa, ja lämpötilakäyrien käsittely nopeuttaa laskenta-aikoja.

PREFACE

This thesis work was carried out to pursue the degree of Master of Science under the supervision of professor Markku Heinisuo from the Faculty of Civil Engineering at Tampere University of Technology and Dr. Jyri Outinen from Ruukki Construction. A big acknowledgement from this work goes to their guidance, as well as Dr. Mauri Laasonen, Dr. Jukka Hietaniemi and Tech.Lic. Dan Pada from the NFD project.

A great deal of the work, especially programming, was done using Ramboll's facilities and the support of their software team Teemu Anttila, Ricardo Farinha and Jyri Leppänen. A special thank also belongs to heads of unit Atte Leppänen and Vesa Hakola for arranging this co-operation.

The work also included a visit to Sheffield University to familiarise with the newest research version of the Vulcan software, and it's developers. I would like to thank professor Ian Burgess for hosting the visit, and Mr. Rui Rui Sun for the valuable guidance.

I thank Ruukki Construction for the financial support of the whole thesis, COST Action TU0904 Integrated Fire Engineering and Responce for the financial support for the Sheffield visit, and The Finnish Association of Consulting Firms SKOL for an additional grant.

Finally I'd like to thank my family and Elina for being there.

Tampere, 14th of March 2012

TABLE OF CONTENTS

SYMBOLS AND ACRONYMS

BIM	Building Information Model
CFD	Computational Fluid Dynamics
E1	The National Building Code of Finland part E1
FDS	Fire Dynamics Simulator, a fire simulation software based on Computational Fluid Dynamics
NFD environment	Natural Fire Design environment, a chain of software tools that support performance-based structural fire safety design
RHR	Rate of heat release
TS	Tekla Structures, a Building Information Modelling software

1. INTRODUCTION

1.1 Background

Design for fire safety of buildings can be based on either prescriptive codes or performance-based requirements. In Finland, as well as in many other countries, both approaches are accepted. While prescriptive codes, based on the standard fire curve defined in [1], have some advantages being simpler and more straightforward, they may sometimes lead to either redundant fire protection of some structures, or even unsafe structures.

Performance-based approaches aim to analyse the real fire scenarios possible for the building in case more carefully. When assuming same temperature curve in every fire in every building leads to uncertain results and sometimes redundant fire protection, analysing the possible fire scenarios gives designer a better understanding of the real fire safety of the structure in case. For this reason, performance-based methods are widely considered preferable approach for fire safety design of buildings. A full scale performance based analysis, however, is time consuming and difficult to perform, and is today used only on special occasions, typically where it is known to lead to cost-effective solutions compared to the prescriptive code approach. Examples of use of the performance-based fire safety design are buildings, whose function requires longer evacuation routes than prescriptive code allows, or steel skeletons of large halls with small fire load, where steel members tend to never reach the temperatures of standard fire curve, and expensive fire protection may not be needed at all.

Performance-based approaches are documented in Natural Fire Safety Concept [2]. To make the methods more useful in normal engineering projects, development has been carried out to combine performance-based fire simulation and structural steel design with building information modelling (BIM) [3], [4], [5]. The integrated design environment described in these papers is called Natural Fire Design environment (NFD environment). NFD environment is a collection of tools, both methods and software, to support performance-based fire design. Development was carried out in co-operation between VTT Technical Research Centre of Finland (VTT), Tampere University of Technology (TUT), Finnish Constructional Steelwork Association (FSCA) and Rautaruukki corporation (Ruukki). Three main results of the project were, as stated in [4], a software combination to use fire simulation software, structural analysis software and building information model together, a guide to the design and documentation for authorities and designers, and a basic data package about typical fire loads and scenarios in buildings [6].

NFD environment, the software combination mentioned above, consists of three programs:

- Tekla Structures version 17.0 (TS): Building information modelling, Commercial software, Tekla Corporation,

- Fire Dynamics Simulator version 5 (FDS): Fire simulation, Open software (Public domain), NIST (National Institute of Standards and Technology, USA), and

- Scia Engineer 2009 (Scia): Structural analysis, Commercial software, Nemetschek Scia.

Data transfer between these programs was implemented by either standard data forms like STEP-files (from TS to Scia), specially programmed macros (from TS to FDS) or combination of both (from FDS to Scia). Programmed macros are property of Rautaruukki corporation and are available to their partner consultant engineers.

As stated in [5], linear finite element analysis gives correct results in structural analysis in fire only, if the structure is statically determined. For statically undetermined structures, non-linear finite element analysis with both geometrical and material non-linearities is needed to get accurate results. Reasons to the need of non-linear analysis are thermal expansion of members and uneven stiffness distribution due to the temperature dependency of elastic modulus of members in different temperatures.

Current design code for steel structures [7] allows resistance checks of individual members as if they were statically determined, i.e. with linear finite element analysis or other simplified method. Code also allows analysis of whole structure or part of a structure, and for these analyses thermal expansion and change of stiffness distribution must be taken into account. Non-linear finite element analysis with both geometrical and material non-linearities is a reliable method for this.

Some software capable of non-linear structural analysis in elevated temperatures is available. Two of them, Vulcan and Safir, were tried out in order to analyse a steel structure in decaying fire by Dan Pada in [8]. Relaying in experience gained, Vulcan was found most suitable non-linear structural analysis program for NFD environment.

In structural design, not only the members, but also connections need to be analysed. Stiffness of connections affects the stress distribution of the whole structure. In case of semi-rigid connections, connection stiffness is temperature dependent.

Analysis method for this kind of structures is presented for 2D analysis in [9] and [10], and for 3D analysis in [11], [12] and [13].

1.2 Purpose of this study

Purpose of this study was to integrate the non-linear structural analysis program Vulcan as a part of the design environment by programming data transfer macros from a building information modelling program Tekla Structures to Vulcan, and from a fire analysis program FDS to Vulcan. A further goal was to speed up calculation time by linearisation and grouping of the time-temperature curves assigned to members in Vulcan.

1.3 Outline of this study

This introductory chapter is followed by chapter 2, which gives more detail on performance-based approach to structural fire safety. Chapter 3 describes the Natural Fire Design environment and programs it consists of, and gives a short example of it's usage. Chapter 4 explains how Vulcan was integrated into NFD environment describing all new components from user's point of view and also briefly explaining how they were programmed. Chapter 5 gives justification to few most critical features of the new solution, including the linearisation and grouping algorithm of temperature curves exported from FDS. Chapter 6 is a case study that uses the new features of the NFD environment, and Chapter 7 includes conclusions of the study together with some insights of future development.

Fire safety is always a sum of multiple things, such as

- Fire resistance of load bearing structures,
- Safe evacuation of occupants,
- Rate of smoke spread,
- Risk of fire spreading to surrounding buildings or spaces and
- Safety of rescue staff.

This thesis concentrates on structural fire safety, but we must bear in mind that mechanical resistance of structures exposed to fire is not the only important factor in fire safety. The generation of smoke, for example, is often more critical to safety of occupants. More complete approaches to Fire Safety Engineering are available in literature.

2. PERFORMANCE-BASED STRUCTURAL FIRE SAFETY DESIGN

2.1 Performance-based fire safety design

Structural fire safety design in most countries has traditionally been based on fire classes, such as the ones given in The National Building Code of Finland part E1 [14], later referred to as E1. These are typically based on the standard fire curve defined in [1], later referred to as ISO-curve. Design procedure with ISO-curve is usually simple and straightforward, but sometimes leads to uneconomically redundant fire protection. Also, sometimes a higher safety level may be needed, than the level gained with design based on ISO-curve.

Natural or performance-based fire safety design is an alternative method for analysing the fire safety of a building design. Method is derived from the Natural Fire Safety Concept [2], that was a European project to develop methods of analysing fire safety more accurately, taking into account factors such as:

- Fire load,
- Ventilation,
- Fire compartment geometry,
- Thermal properties of the enclosing structure,
- Active fire fighting measures, and
- Probability of fire activation

Today many countries accept performance-based fire safety design as either a primary or an alternative method of fire safety design. The National Building Code of Finland part E1 ([14], 1.2), for example, states an essential requirement, which includes, that

- The load bearing structures shall sustain for the minimum time,
- The generation and spread of fire and smoke shall be limited,
- The spread of fire to neighbouring buildings shall be limited,

- The occupants shall be able to leave the building or be rescued by other means, and
- The safety of rescue teams in building shall be taken into consideration.

E1 ([14], 1.3) gives two alternative ways to satisfy the essential requirement:

1. Using the fire classes and numeric values given in E1, or
2. Designing the building based on design fire scenarios.

Method 1 remains the most common way of design, because it is a relatively simple procedure that normally leads to safe structures. Design fire scenarios are equally acceptable design approach. Today they are used mainly in special cases where E1 tables are known to lead to non-practical solutions, for example uneconomically redundant fire protection of steel structures. More detailed and time consuming fire analysis may be worth the effort, if the need for extra fire protection can be reduced.

2.2 The design procedure using performance-based approach

The two alternatives of fire safety design given in E1 [14], the one based on fire classes and numeric values of E1 and the one based on design fire scenarios, were described above. A practical approach to Fire Safety Design in accordance with E1 given in [15] introduces three alternative methods, dividing the use of design fire scenarios in two alternative ways:

1. Using the fire classes and numeric values given in E1,
2. Not using values of E1 as they are, but using calculations to verify that the requirements of E1 are fulfilled by other means. This method is useful when only few, clearly defined exceptions to E1:s prescriptive guidelines are made
3. Analysing the whole building with probablistic risk assessment and comparing the risk level to an acceptance criterion.

The first method is the same as above, and is described in detail in [16]. The second and the third one are two different approaches using design fire scenarios.

Figure 2.1 illustrates the design procedure as it was described in Natural Fire Safety Concept. Design fire characteristics are determined and analysed carefully using validated methods, and the realistic design fire curve is then used to determine behaviour of the structure in fire. Realistic evacuation times can also be determined and compared with fire resistance time of structures. However, because performance based fire models normally do not last infinitely, it is possible to set the assessment criteria so that structure must sustain the whole fire, i.e. minimum fire resistance time is infinite.

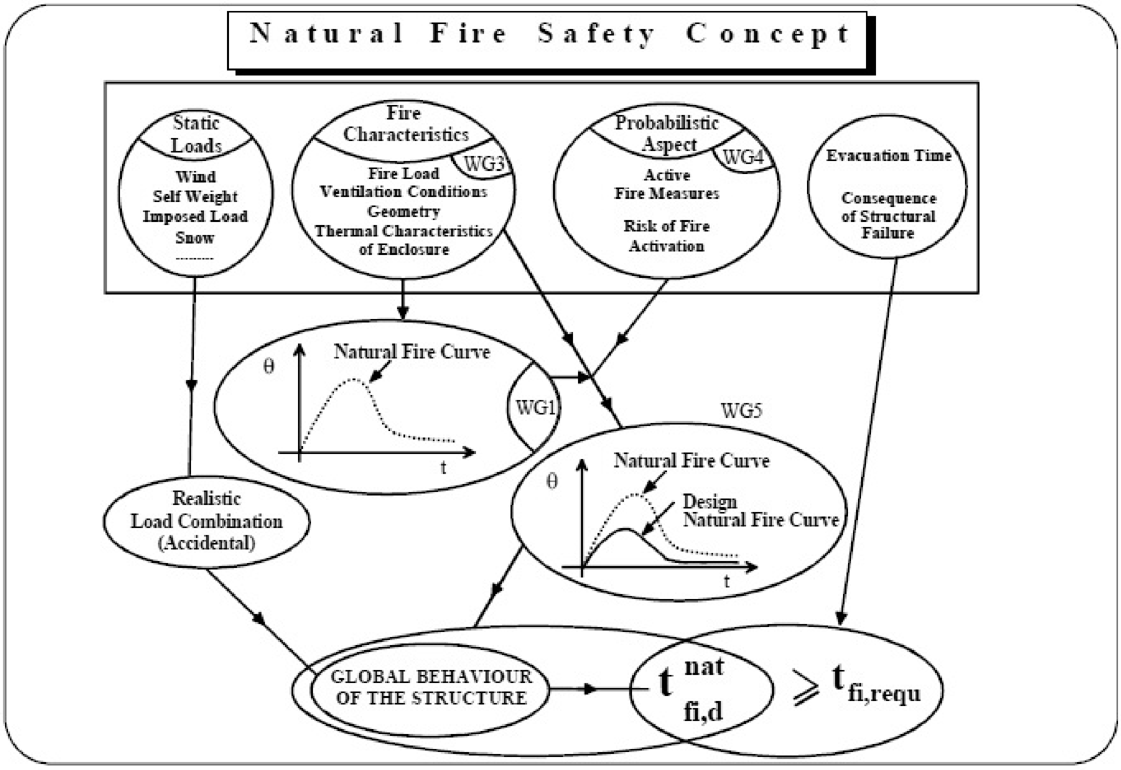

Figure 2.1: The design procedure using natural fire design [2]

2.3 Methods of fire development calculation

2.3.1 Fire development models

When using design fire scenarios for design, temperature curves are determined from design fires using a fire development calculation model. There are different development models that can be used for different purposes. Models may be divided in four categories:

- Parametric models
- Localised fire models
- Zone fire models
- Field models

Parametric models, such as the one in EN1991-1-2 Appendix A [17] are relatively simple models for post flashover fire behaviour. They take into account the amount of fire load, size of the fire compartment and it's openings. An example of a parametric temperature curve is presented in figure 2.8.

Localised fire models are used to analyse a local fire before flashover or when flashover does not happen. Localised models are used to determine temperature within and above the flames, and they are useful when the highest possible temperature of a structure is needed, which usually is the case in structural fire engineering.

Zone models divide space in one or more zones that are assumed to have uniform temperature. Due to the assumption of uniform temperature, zone models cannot be used to determine the highest temperature directly above the flame. Typically they are used in analysis of evacuation safety (all occupants must be evacuated before the thickness of relatively safe lower zone reduces dangerously near evacuation routes) or to predict whether flashover will happen or not.

Parametric, localised and zone models are all based on simplifications that make them unreliable when used for geometrically challenging fire compartments. When that is the case, sophisticated field models are needed. Field models are based on Computational Fluid Dynamics (CFD). Fire compartment is divided in a grid of nodes, and the differential equations of the gas properties in each node are solved numerically on a computer. Calculation is rather time consuming. Using denser grid gives more accurate results, but slows the computation significantly. As computational power increases, field models are becoming more common in all fire engineering. The most commonly used software in engineering projects is Fire Dynamics Simulator (FDS), which will be discussed in more detail later in this study.

2.3.2 Design fires

All fire development models aim to determine the effects of a design fire in a specific building or fire compartment. Parameters needed may be divided in three categories: compartment characteristics, design fire characteristics and design fire placement. Compartment characteristics are information of the geometry, material thermal properties and ventilation conditions in fire compartment. Design fire placement should be selected so that the most dangerous case will be checked. In evacuation simulations, for example, most dangerous case is often when design fire blocks one of emergency exits.

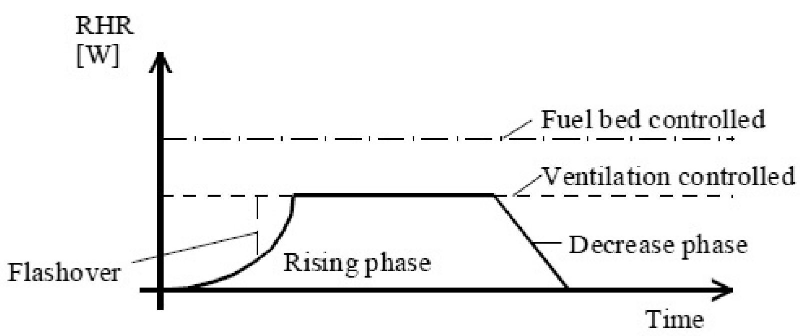

Figure 2.2: Example of RHR curve [2]

Design fire characteristics can be expressed in a form of a Rate of Heat Release (RHR) curve. RHR curve defines how fast the energy stored in fire load will release in fire. Figure 2.2 is an example of an RHR curve used in design. These design curves are approximations of real fire RHR developments. The shape of the design RHR curve depends on the fire load, fire growth rate, whether the fire is fuel or ventilation controlled, and whether flashover will happen or not. Details of design RHR curve are discussed in more detail for example in [2] and determining fire load and fire growth rate in [17].

Determining reliable design RHR curves may be difficult, especially for specific localised fires for which statistical fire load data is not available. There are reviews on experimentally and analytically determined design fires in literature [6].

2.3.3 Examples of fire development models used in structural fire engineering

Different fire development models are used for different purposes in fire safety design. In structural fire engineering the highest possible temperature of a structural member is usually needed. For post-flashover fires parametric or field models are used. For pre-flashover fires either localised fire models or field models are used, with the design fire placed directly below the critical member.

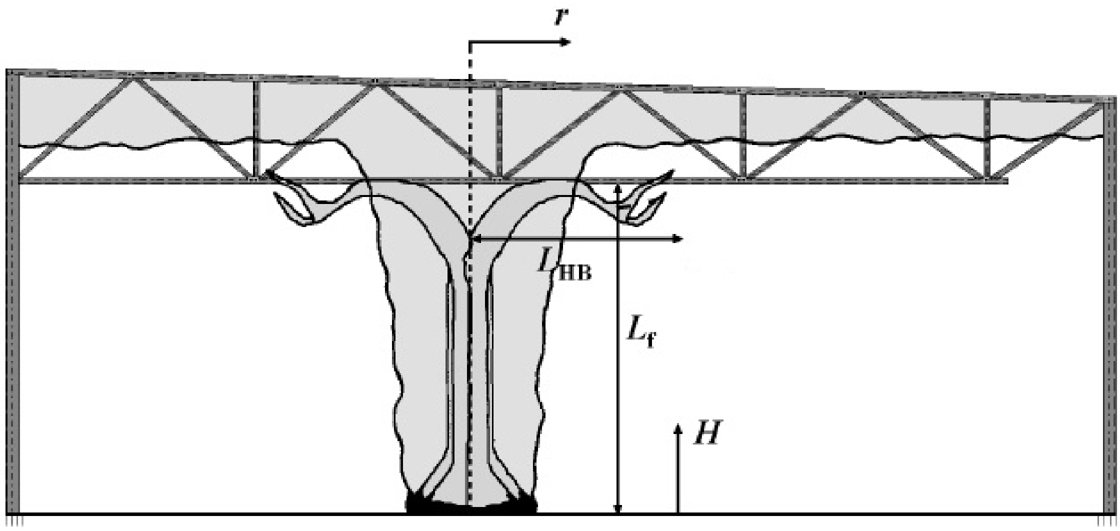

Figure 2.3: Localised fire below a roof truss [18]

In his M.Sc. thesis Junnonen [18] used an approach where four different simplified methods were used together in a scenario with a local fire below a roof truss. Flame height was calculated with Heskstad's model. For members touched by flames Hasemi's model was used to determine member temperature. Heskestad's and Hasemi's models are included in EN1991-1-2 appendix C [17]. For members

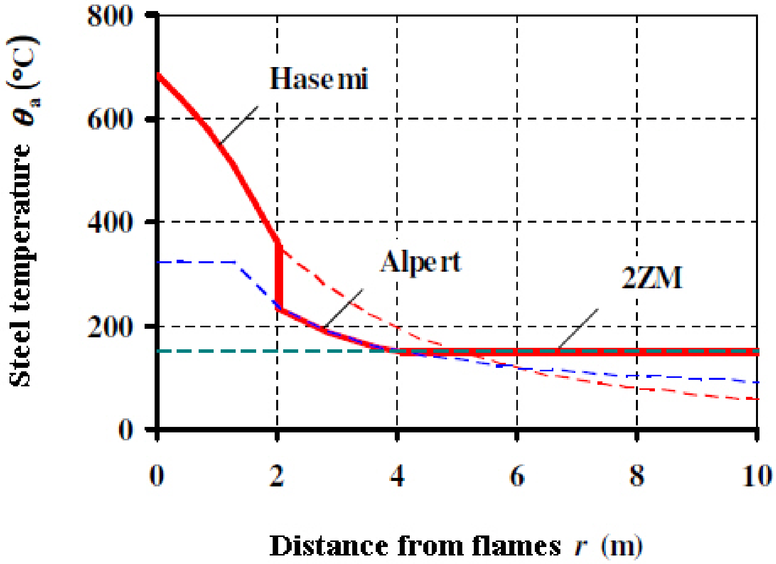

Figure 2.4: Temperatures of lower vertical members of truss [18]

near the flames but not touched by them another localised model, Alpert's model [19] was used. For the rest of the members, a 2-zone model was used. Figure 2.3 shows Junnonen's case and figure 2.4 the temperature of the lower vertical members of the truss as a function of distance from the flame centre line.

In a case study published in [20] a simplier approach was used. There each member was checked assuming a localised fire directly below the member, so that 2-zone model and Alper's model used by Junnonen were not needed. The localised model used was McCaffrey's model [21].

As stated above, field models are the most sophisticated method of fire development analysis. They do not have the limitations of other models, and can be used when enough know-how and computational power is available. FDS is the most commonly used software. A published case study of a sports centre built in Helsinki used FDS simulations to determine design temperatures for steel roof trusses [22]. Figure 2.5 shows a restaurant coat rack fire of the project.

2.4 Structural fire resistance of steel structures

Fire weakens practically all construction materials. Steel does not burn, but it's strength and elastic modulus reduce as temperature rises. Steel material properties used in fire design are given in detail in Eurocode 3 part 1-2 chapter 3 [7]

Essential requirement for load bearing structures is, that the action of design values of imposed loads is equal or smaller than the corresponding design value of resistance in fire. Fire resistance check is an accidental limit state check, so both loads and material properties have smaller safety factors, typically 1.0.

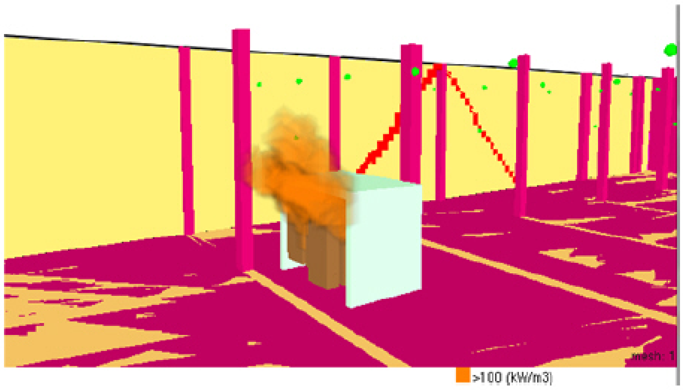

Figure 2.5: A coat rack fire simulation in FDS [22]

Eurocode [7] states three ways of verifying fire resistance of structure: simplified calculation models, advanced calculation models and testing. If calculation methods are used, analysis may be done for individual members, parts of structure or the whole structure.

The simplest and most common method is to analyse individual members using simplified calculation models. This approach ignores effect of thermal expansion and re-distribution of actions due to change of stiffness of the structure, but is simple to perform and gives correct results for statically determined structures. Aim of this study is to use Vulcan software to analyse whole structure or parts of structure using advanced mehtods, but let's first take a look at the simplified methods of Eurocode 3.

2.4.1 Simplified calculation models of Eurocode 3

A typical design assignment in structural fire engineering is such, that dimensions of steel sections have already been determined in structural design in service temperatures, and aim of fire design is to determine the required fire protection.

When using simplified calculation methods for individual members, design procedure illustrated in flowchart in figure 2.6 is efficient, because design temperature $\theta_{a,d}$ only needs to be determined once, and then compared with several maximum steel temperatures $\theta_{a,max}$ calculated using different fire protection alternatives.

If analysis is not done on individual members but either on a whole structure or a part of structure, design procedure of figure 2.7 needs to be used, because steel temperature distribution affects also the mechanical actions $X_{E,d}$ and not only the resistance of members $X_{R,d}$. This approach requires more calculations, if multiple fire protection alternatives are compared.

Instead of determining design temperature of member, a simplified method may be used to determine so-called critical temperature $\theta_{a,cr}$. Critical temperature only

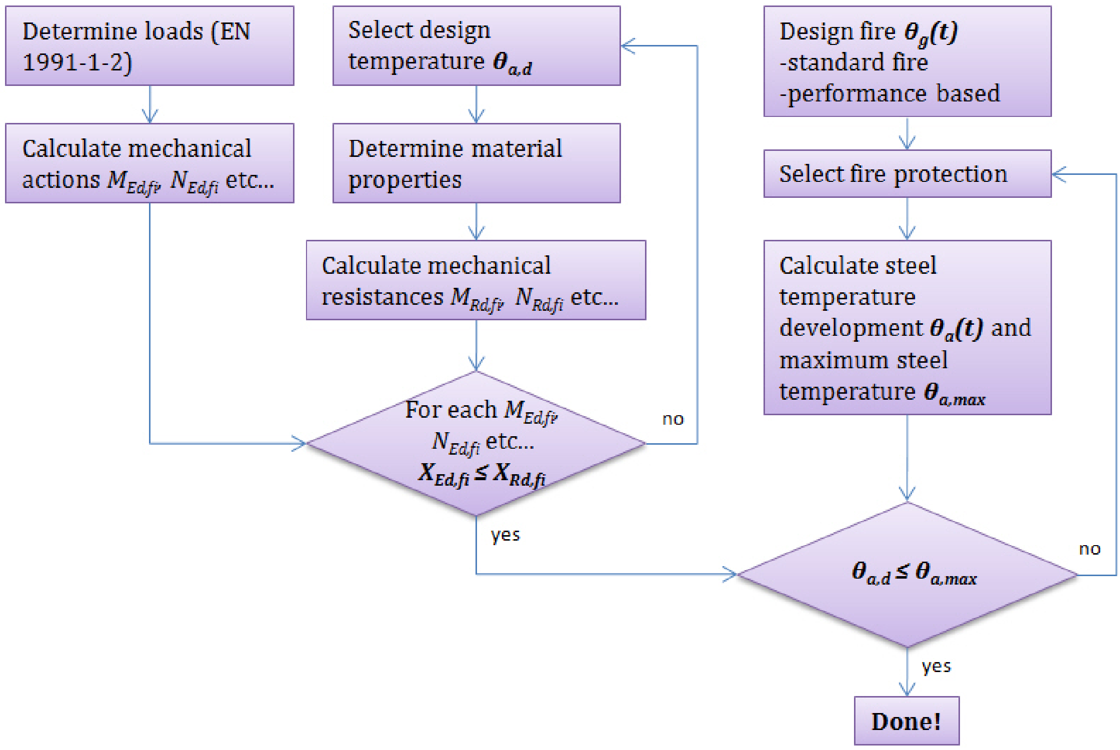

Figure 2.6: Structural steel fire design procedure, where design temperatures are compared with maximum temperatures

takes into account the reduction of the yield limit at elevated temperatures, so it may not be used for members that need stability checks. Stability depends on both yield limit and elastic modulus of steel. Because buckling and torsional lateral buckling are often critical to members, critical temperatures are not used for design in this study.

2.4.2 Steel temperature development

Calculation method for steel temperature development, when temperature development of surrounding gas is known, is given in Eurocode 3 part 1-2 [7] section 4.2.5. Surrounding gas temperature may follow the standard fire curve or some performance based fire development. Eurocode formulas take into account the section factor A_m/V of steel member and properties of fire protection material. Temperature of an unprotected steel section will follow the gas temperature with a short delay, and for protected steel section the delay is longer. With decaying fire models the steel temperature will start reducing when gas has cooled down colder than steel. Figure 2.8 shows and example of temperature development of a typical protected and unprotected steel section in parametric fire.

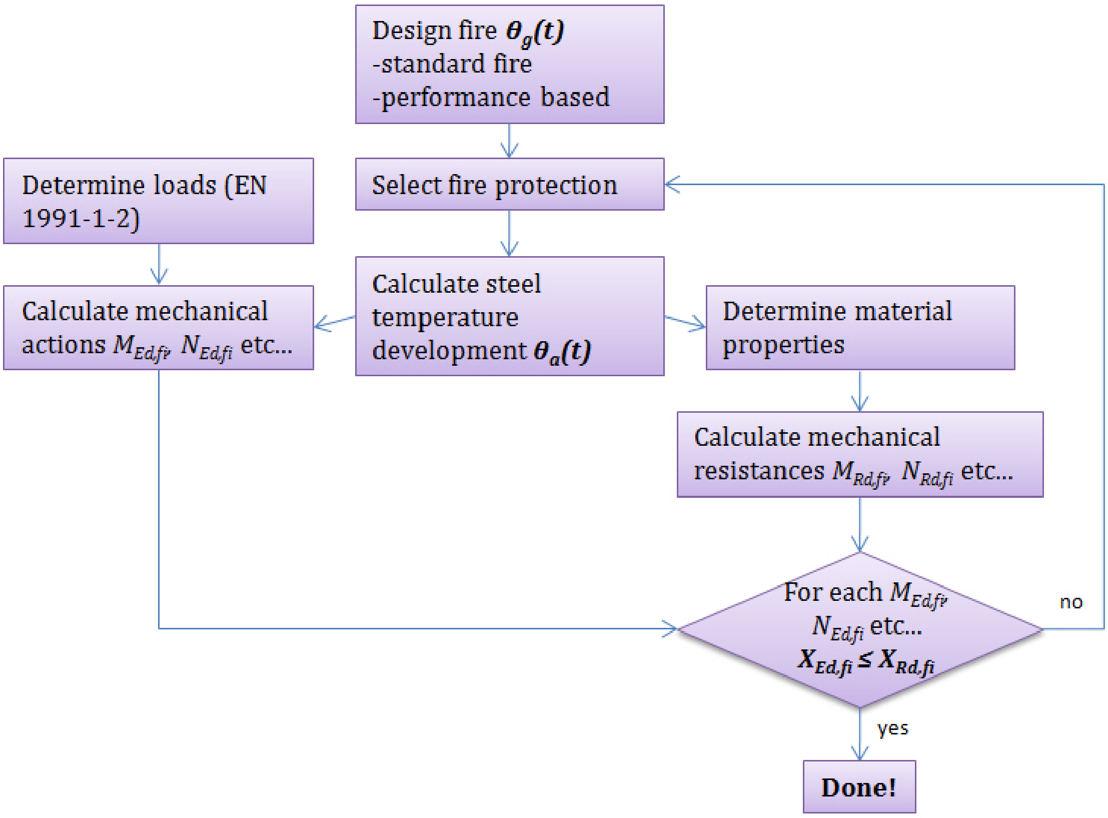

Figure 2.7: Structural steel fire design procedure, where steel temperature is calculated before structural analysis

2.4.3 Mechanical resistance

Mechanical resistance of steel members can be calculated by reducing the material properties, yield strength and sometimes elastic modulus, according to the temperature. When using design procedure of figure 2.7, steel temperature is already known. When using design procedure of figure 2.6, design steel temperature must be calculated either by making an initial quess and then iterating the equations of resistances until the highest allowable temperature is found, or by solving the temperature from the equations by setting the resistances equal to actions of design loads. Figure 2.6 illustrates the iterative alternative, for which design code formulas can be used as they are presented in the code.

Equations of mechanical resistances of steel members are presented in Eurocode 3 part 1-2 section 4.2.3 [7]. Different rules apply to

- Members subject to tension,
- Members subject to compression,
- Beams subject to bending, and

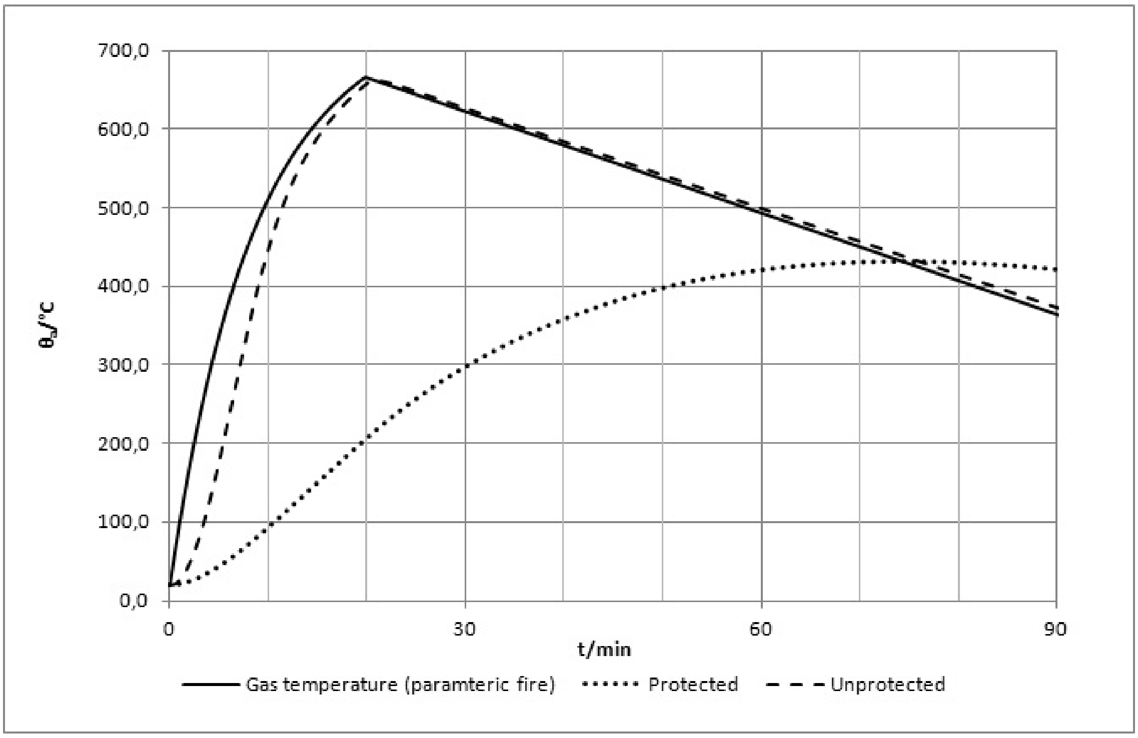

Figure 2.8: Temperature development of protected and unprotected steel section in parametric fire

- Members subject to combined bending and axial compression.

Tension members are checked against axial tension, and compression members against buckling. Beams are checked against bending, lateral torsional buckling and shear. Members subject to combined actions are checked with formulas that take into account buckling and lateral torsional buckling.

Material properties are reduced as a function of temperature according to EN 1993-1-2 chapter 3. Accidental limit state safety factors are used. For buckling lengths, less conservative values than in service temperatures may be used under certain conditions, see EN 1993-1-2 4.2.3.2 [7] and Teräsnormikortti 13 [23].

2.4.4 Structural resistance and deformation criteria in testing and advanced calculations

Different equations of mechanical resistance of different members presented above are needed in Eurocode's simplified calculations, because calculations expect a linear material model for steel. Advanced calculations with non-linear material model are more difficult to perform, but the resistance criteria for statically determined structures becomes simple - a so called run-a-way deformation failure occurs at time t_f of equation 2.1, see figure 2.9.

$$\lim_{t->t_f} (dv/dt) = \infty, \tag{2.1}$$

where t is time, t_f is the time of failure and v is deflection of a member.

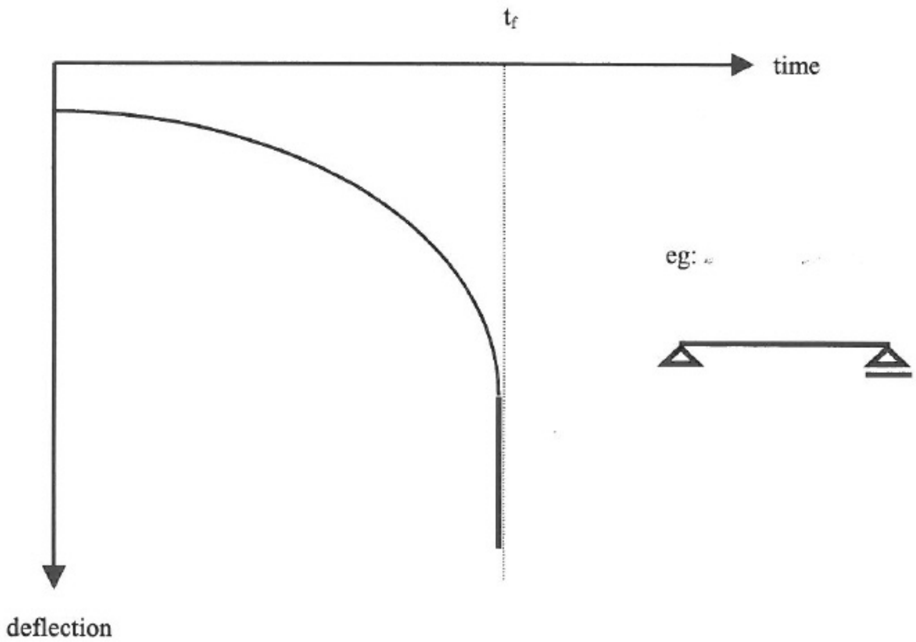

Figure 2.9: Run-a-way deflection of a simply supported beam in a fire [24]

Run-a-way failure does not take place in all cases. For example a beam with fixed supports develops membrane forces to replace the reduced bending capacity, see figure 2.10. The beam may reach extremely large deflections without breaching the failure condition 2.1. Such large deformations may often be unacceptable, for example if deformation damages the fire protection of the beam. Hence a deformation limit 2.2 is set.

$$v_{rel} = L^2/400h, \tag{2.2}$$

where v_{rel} deflection relative to the supports, L is the span and h is the depth of the member.

In some sources criteria 2.2 is simplified to form 2.3, where it is assumed that member dimensions have a ratio $L/h \approx 20$, which is common for hot rolled steel beams.

$$v_{rel} = L/20. \tag{2.3}$$

Vulcan, which is the structural analysis software in the focus of this study, has failure criterion 2.1 hard coded in it. Program's solver simply fails to find equilirium as the deflection reaches infinity, and the analysis stops at the time of failure.

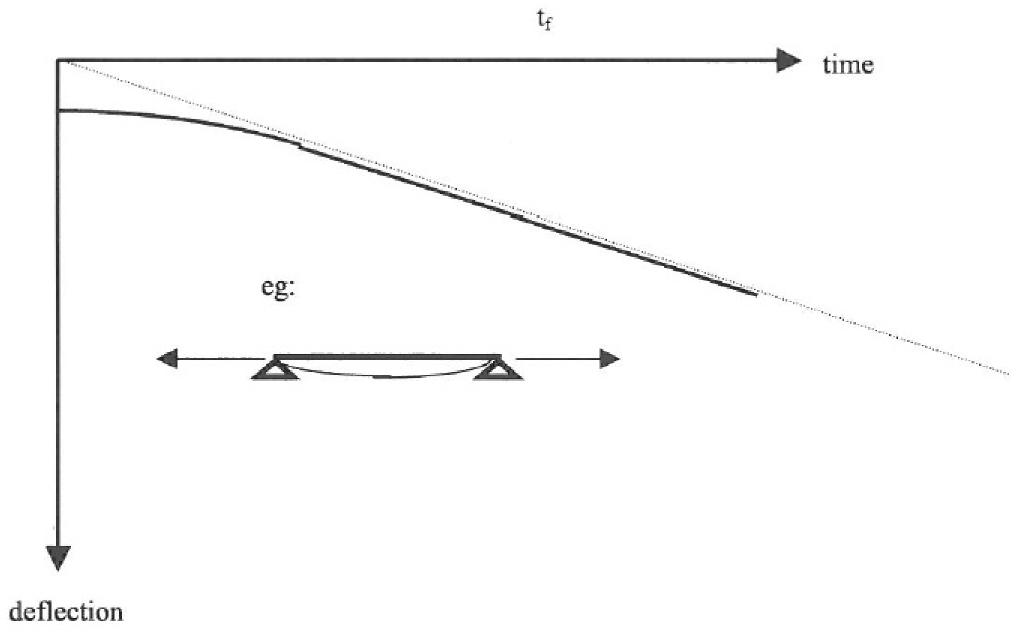

Figure 2.10: Deflection of a beam with membrane action in fire [24]

Separate checks for criterion 2.2 need to be performed by the designer when using Vulcan. Possibility to automatise this is studied later in this study.

Criterion 2.1 also covers stability checks needed for the analysis. Local or global buckling will cause run-a-way deflections of members. Results, however, depend on initial eccentricities of the members. Eccentricities must be modelled in Vulcan manually. Correct procedure would be finding out first buckling modes of the structure, and scale the nodal displacements of the mode to fit the manufacturing tolerances specified for example in standard EN1090 [25]. Automatic setting of these displacements could not be implemented in this study, so manual setting will still be needed in the future when accurate stability analysis is carried out.

3. NATURAL FIRE DESIGN ENVIRONMENT

3.1 Natural Fire Design environment before this project

The Natural Fire Design environment has been developed since 2007. It's status at the beginning of this project was roughly the same as described in [4]. Figure 3.1 illustrates the design process when using NFD environment.

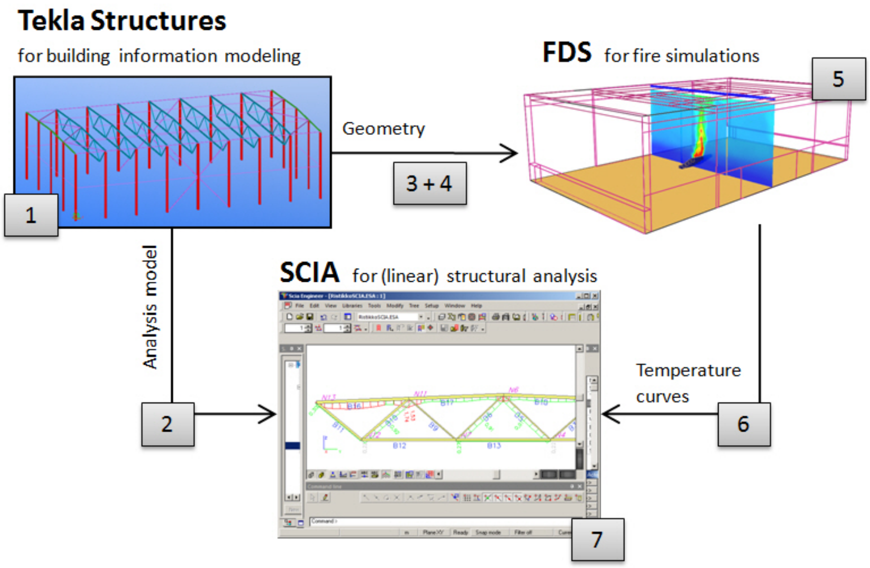

Figure 3.1: Design process when using NFD-environment.

Process stages are numbered in accordance with [4]:

1. Modelling the frame of the building by structural engineer in Tekla
2. Exporting the frame model to structural analysis software Scia by structural engineer
3. Definig the fire scenario(s) together with fire consultant and authorities. Modelling the fire scenario in Tekla and exporting to FDS.
4. Definig the temperature points needed for structural analysis in elevated temperature. Exported from Scia in XML-format and read to FDS.
5. Fire simulation in FDS

6. Reading in the temperature curves to Scia

7. Dimensioning the structures in fire temperatures, adding fire protection if needed.

3.2 Integration of Vulcan into NFD environment

The core objective of this study was to integrate Vulcan structural analysis software into NFD-environment. This was done by programming a data transfer link between Tekla and Vulcan (2B in figure 3.2). Also the existing link between Tekla and FDS was modified to support reading the temperature point co-ordinates of each beam from Vulcan files as well as from Scia xml-files (4B in figure 3.2), and writing the temperature curves in Vulcan format (6B in figure 3.2).

Figure 3.2 illustrates the design process enhanced with Vulcan for non-linear Finite Element Analysis.

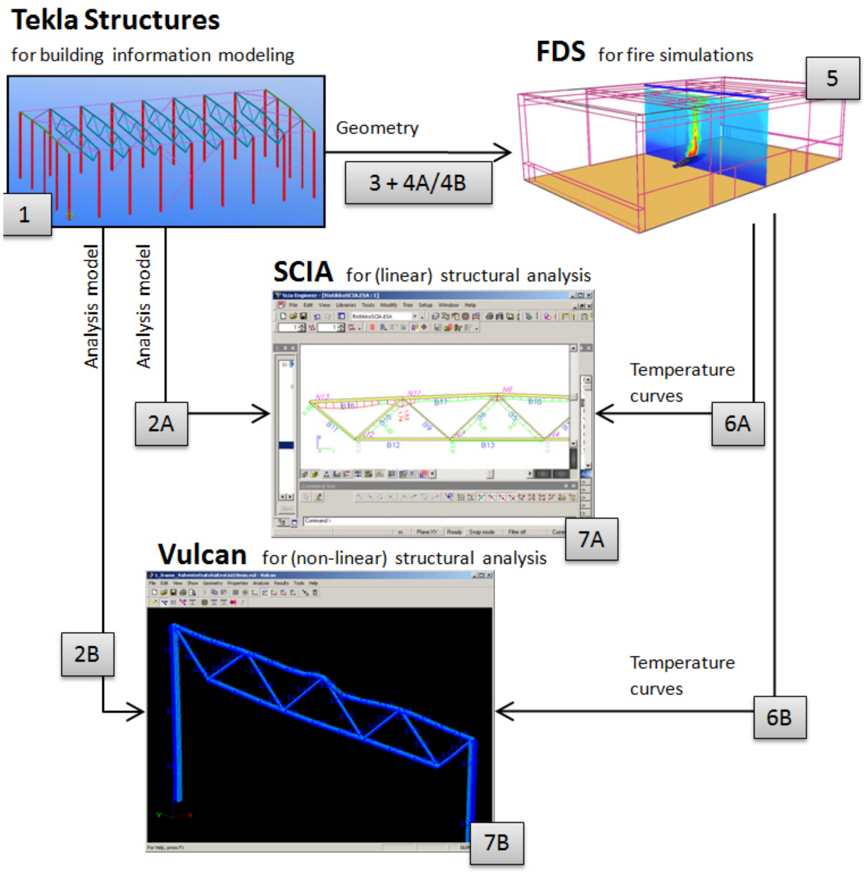

Figure 3.2: Design process when using NFD-environment enhanced with Vulcan.

3.3 Programs used in NFD environment

3.3.1 Tekla Structures

Tekla Structures (TS) is a Building Information Modelling software widely used by structural engineers in Finland and worldwide. TS is commercial software with closed file formats, but interoperability with some open data formats and, more importatly, the Tekla Open API (application programming interface) make it suitable platform for third party design programs such as the NFD tools. NFD currently (2011) uses TS version 17.0.

3.3.2 Fire Dynamics Simulator

Fire Dynamics Simulator (FDS) is a fire simulation software based on Computational Fluid Mechanics. It is open software (public domain) developed by NIST (National Institute of Standards and Technology, USA). FDS is widely used and well validated for fire simulations of buildings. FDS version 5 was found most suitable for NFD environment.

FDS itself does not have any graphical user interface. It reads input from text files and prints output to other text files. A visualisation program Smokeview (SMV), also by NIST, is used together with FDS to visualise results.

3.3.3 Scia Engineer

Structural analysis in NFD environment is done by Scia Engineer 2009 program. Scia is a Finite Element Analysis software for structural engineering. It includes beam and plate elements in 2D and 3D. Scia's own file format is closed, but it supports a variety of import/export formats. Ability to import user defined time-temperature curves for beams in fire was the key feature for which Scia was selected to be used with NFD. Many other commercial programs only support ISO-curve and some other pre-defined time-temperature curves.

3.3.4 Vulcan

When linear analysis is not sufficient, non-linear finite element software with both geometry and material non-linearities is needed. There are several advanced multi-purpose FE-codes available, LUSAS, Ansys and Abaqus just to mention a few. These are capable of almost any kind of analysis with the right pre-processing, but tend to be too expensive and complicated to use for everyday structural engineering. In [8], two programs specially developed for analysing structures in fire, SAFIR and Vulcan, were tried in analysing a whole building in 3D with beam and truss elements.

Vulcan was found more suitable for the purpose, and is therefore to be integrated in NFD environment.

Vulcan is originally based on Instaf-program for two-dimensional non-linear finite element analysis. Instaf was developed in University of Alberta in 1980. It was then developed further in University of Sheffield for analysing structures in fire. 1997 the program was re-named as Vulcan to emphasise the difference from original Instaf-program.

Vulcan currently has two completely different versions, a research version written in Fortran77 like Instaf was, and a completely re-programmed commercial version written in C++. The commercial version is available for industry via a University of Sheffield spin-off company Vulcan Solutions. In this study, the commercial version was integrated in Ruukki's NFD-environment. A study of the possibility of integrating also the research version is in section 5.4.

The commercial version deals with steel beam and concrete slab elements in 3D with geometrical and material non-linearity. Stress-strain curves and thermal expansion properties are hard coded in the material models of concrete and steel. Cross sections of the elements are divided into segments to allow temperature, stress and strain to vary within the cross section. Vulcan is well validated against test data, including large scale fire tests at Cardington, UK. [26]

3.4 Data transfer in NFD environment

Figures 3.1 and 3.2 above illustrated the programs of NFD environment, but did not explain much of the data transfer between them. Figures 3.3 and 3.4 give more detail in that. The programming interface Tekla Open API allows third party functionality to be added on the building information modeling software. Programmed Open API macros for running FDS simulations and exporting the analysis model in either Scia or Vulcan is the way NFD tools are integrated in Tekla.

There are three data transfer macros in NFD environment. Some other macros are also used for modelling the fire scenarios and other model objects needed by FDS. Below we take a look at the three data transfer macros.

3.4.1 FDS model from Tekla - FDS adapter macro

Macro FDS adapter is the core part of NFD environmet. It handles all data transfer in and out of FDS. It's main purpose is to write FDS input file and start FDS, but it also reads FDS results after the simulation and writes them to Scia or Vulcan input files.

FDS input is an ASCII file, that defines the computational mesh, geometry, design fire and other things needed for FDS simulation. If time-temperature curves

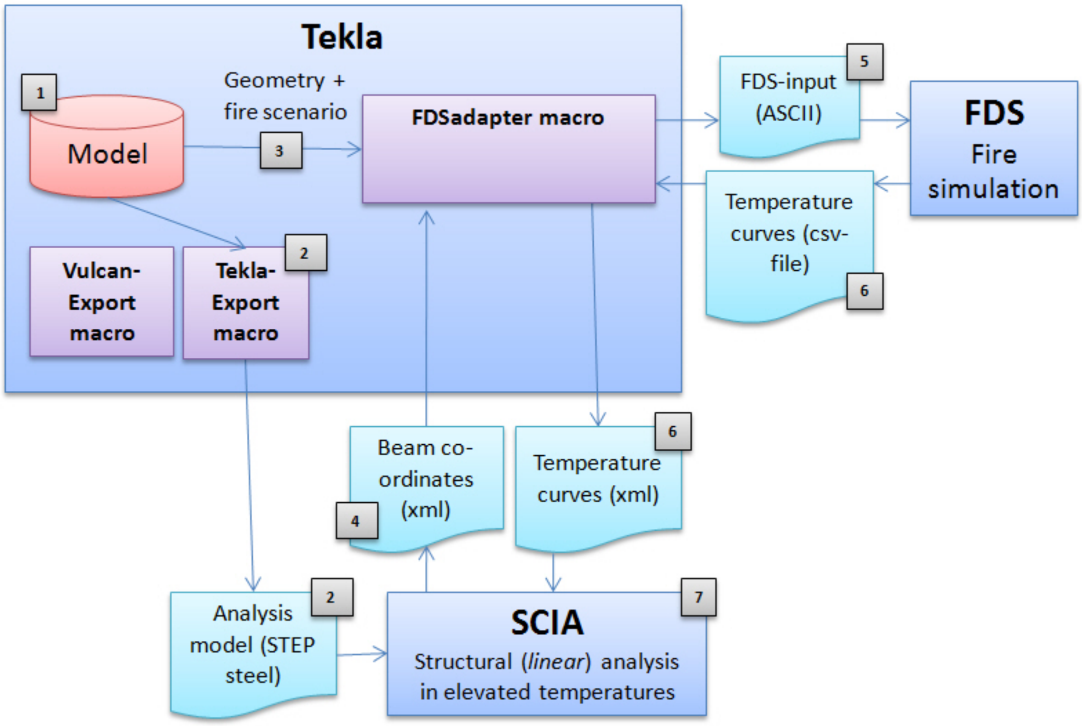

Figure 3.3: Data transfer in NFD environment, when Scia is used.

of specific points (beam mid points in FEA) are needed, the point co-ordinates need to be defined in FDS input file before simulation. For this, FDS adapter needs to read data from the analysis model. From Scia analysis model, geometry needs to be exported in xml-format after the step-file is imported in Scia. From Vulcan, FDS adapter can read the Vulcan input file (.vul) directly. In both cases, the analysis model must be created before FDS simulation to get the temperature points, but for solving the analysis model in fire temperatures, FDS simulation must be completed.

FDS outputs temperature curves of the pre-defined temperature points in csv-file that can be opened for example in MS Excel. Scia and Vulcan need the temperature data in their own format, xml for Scia and ASCII for Vulcan, so FDS adapter needs to process the temperature data after FDS simulation is finished. As a part of this study, a functionality is added in FDS adapter to also linearise and group the temperature curves so that multiple members of the analysis model may have the same, linearised temperature curve. This is expected to speed up calculation time in structural analysis. Grouping and linearisation is discussed in more detail in section 5.1 of this study.

3.4.2 Scia model from Tekla - Tekla Export macro

Data transfer between Scia and BIM-software is discussed in [27]. Methods can be divided in three levels:

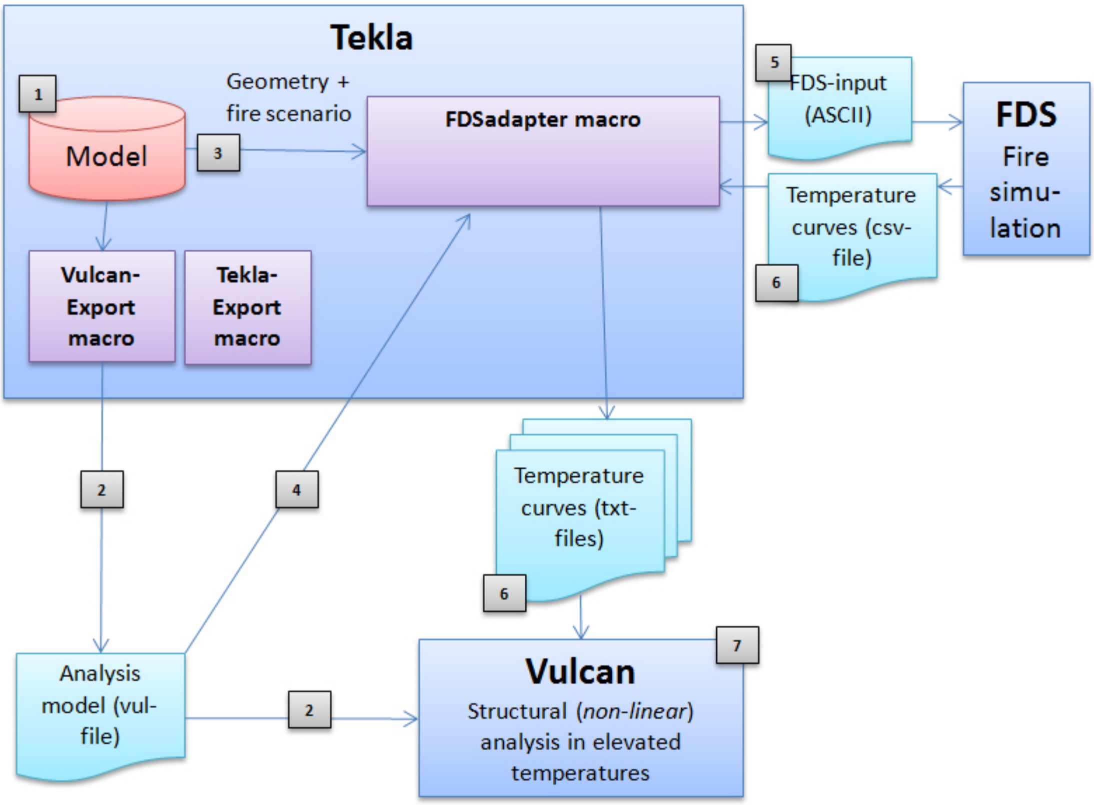

Figure 3.4: Data transfer in NFD environment, when Vulcan is used.

1. Importing/exporting data, dwg-files for example,
2. Using standard exchange formats such as IFC or CIS/2, and
3. Using direct links between the programs.

According to [27] there exist an external direct link plug-in to be used between Scia and Tekla, but attempts to use it in NFD environment have failed. This kind of problems are not rare in BIM/CAD/FEA interoperability issues, mainly due to the version dependency with all three programs: the source model, the target software and the external plug-in.

However, both Tekla and Scia support a variety of standard exchange formats, and a post-processor macro Tekla Export has been developed for the Tekla-Scia transfer with STEPSTEEL-format files. STEPSTEEL files exported from Tekla need post-processing, because Scia uses different naming system for standard profiles.

It should be noted, that Tekla Export macro does not export the Tekla analysis model created in Tekla from Analysis & Design Models -dialog, but a simple geometry model of the selected parts. This way, no information of loads, load combinations, boundary conditions or beam end degrees of freedom is exported. This was the only way to export analysis model when Tekla Export was first developed, but programming of Vulcan Export revealed a spectacular development in Tekla Open API analysis properties during last few years, and today analysis model could be completely created in Tekla Structures and exported to Scia just for solving. This approach has the advantage of storing all the data in Tekla model for better consistency.

3.4.3 Vulcan model from Tekla - Vulcan Export macro

VulcanExport macro was the key product of this study. It's purpose is to export analysis model from Tekla to Vulcan for analysis. Vulcan only accpets it's own input file (.vul), so Vulcan Export macro needs to read the analysis model from Tekla database using Tekla Open API and then write the data in Vulcan format. Vulcan file is Fortran-style ASCII code readable for human, so the functionality was not that difficult to program.

Vulcan Export expects user to first create an analysis model in Tekla using Analysis & Design Models tool from Tekla menu. One Tekla model may have several analysis models, and one of them can be exported to Vulcan at a time. Vulcan does not have a concept of load combinations, so user must select only one load combination to be exported. For multiple load combinations, multiple Vulcan models are needed so that each has only one load combination.

4. SOFTWARE DEVELOPMENT OF TOOLS TO INTEGRATE VULCAN IN THE NATURAL FIRE DESIGN ENVIRONMENT

Section 3.4 descirbed what kind of data macros Vulcan Export and FDS adapter process and why. This chapter will concentrate on how to use them and how they were programmed. In other words, what needed to be done in order to integrate Vulcan into NFD environment, and also what does the user need to do to be able to use Vulcan together with other NFD tools.

Macros were programmed in C# programming language using Tekla Open API (Application Programming Interface). Tekla Open API provides an interface for third party applications to interact with Tekla model [28]. These Open API applications may be used for purposes of

- automating routine tasks,
- integrating Tekla Structures with user's process or other software, or
- developing additional functionality.

4.1 Vulcan Export macro

4.1.1 Usage and functionality

Before Vulcan Export macro can be used, an analysis model must be first created in Tekla Analysis & Design Models dialog, see figure 4.1. Model should have some parts modelled. Also loads should be modelled, unless it is decided to model loads afterwards in Vulcan. New analysis model is created from the 'New' button on Analysis & Design Models dialog. User needs to give some attributes regarding analysis model creation, most important being analysis model name, creation method ('By selected parts and loads' is probably most common selection), analysis application (this selection is overridden by Vulcan Export, so it has no effect) and member axis location behind 'More settings' button.

After analysis model is created, user still needs to create load combinations for it by clicking 'Load combinations' button on Analysis & Design Models dialog when the created analysis model is selected. Combinations can be added either one by one

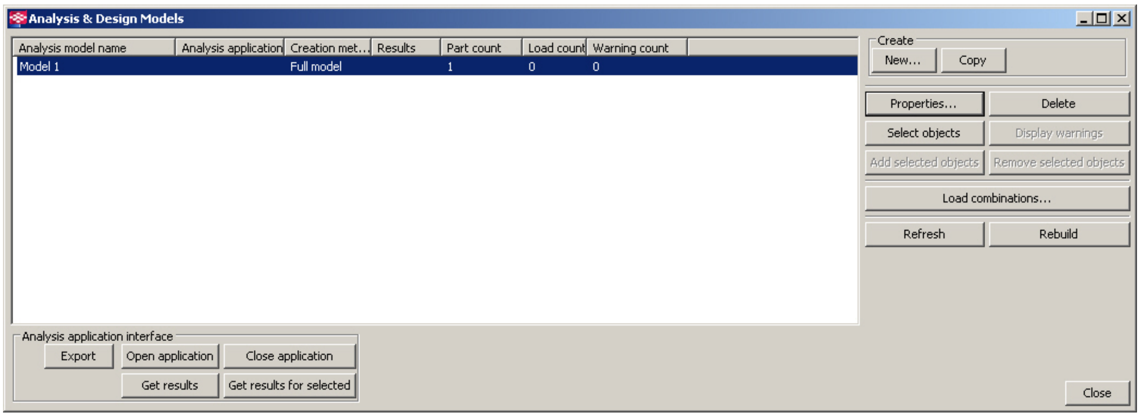

Figure 4.1: Analysis & Design Models dialog in Tekla Structures.

from 'New' button, or automatically from 'Generate' button. Like on other Tekla user interface dialogs, user must click 'Apply' to save the load combinations -just clicking 'OK' does not modify model.

Analysis & Design Models dialog in Tekla has an "Export" -button, but that button cannot be used with Vulcan because it is not accessible from Tekla Open API programming interface. Instead, after the analysis model is created, user must start Vulcan Export macro either from Tekla's Macros dialog or from a toolbar, if Vulcan icon has already been set on a custom toolbar, see figure 4.2.

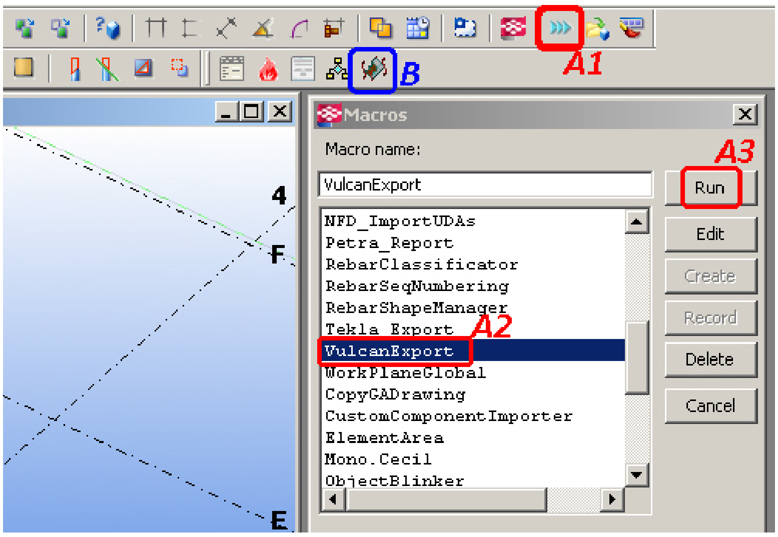

Figure 4.2: Opening Vulcan Export macro from either A: Macros dialog, or B: Tekla toolbar, if toolbar shortcut has been created.

When Vulcan Export window (figure 4.3) opens, user needs to define four settings. From "Select analysis model to export" drop down menu, the analysis model wanted to be exported is selected. Drop down menu shows all the analysis models that were defined in the model when Vulcan Export was opened. If user creates new analysis models while Vulcan Export is already open, a click of "Refresh" button next to the

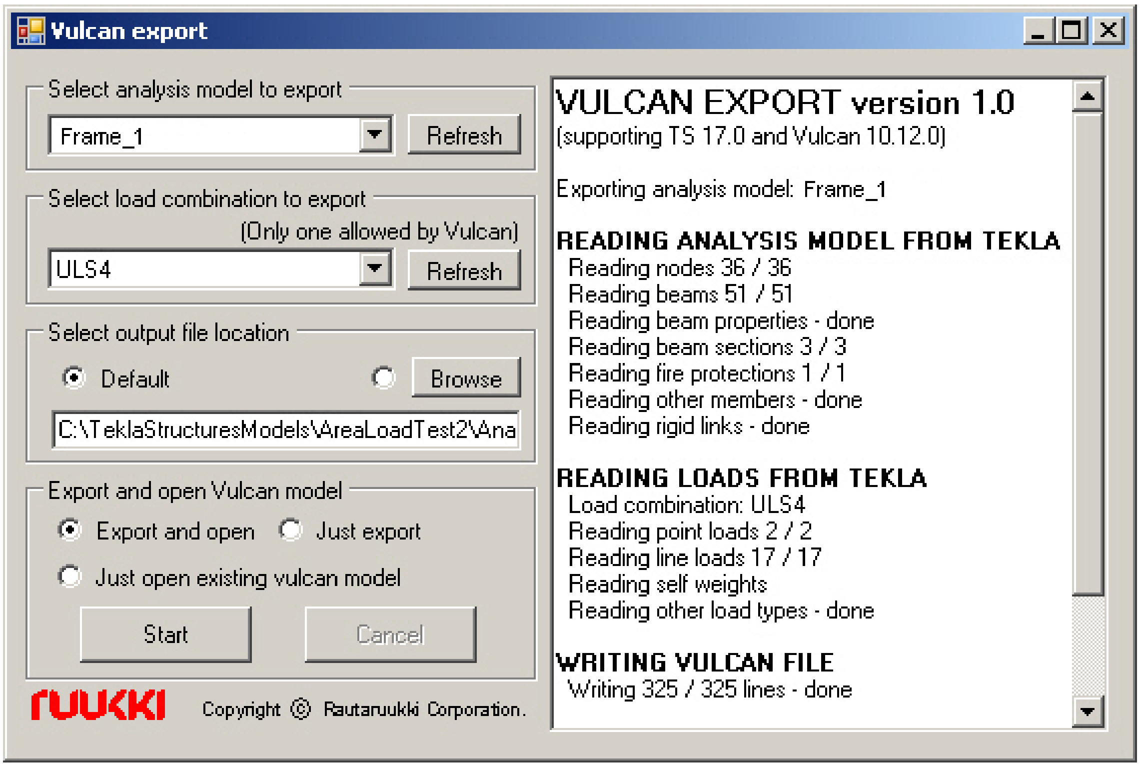

Figure 4.3: VulcanExport dialog.

drop down menu is needed for the new analysis models to be shown.

Another drop down menu is used to select the load combination to be exported. Vulcan can only have one load combination it one file, so user must select which one to export. Third option to be selected is location of the output file. Vulcan Export macro exports a Vulcan format (.vul) text file, which then can be opened in Vulcan. Default file location is under Tekla model folder in *Analysis/Analysis_model_name* sub-folder. User may specify any other file location too. Last selection is, whether Analysis model is exported and opened in Vulcan, just exported, or just opened (if it has been previously exported.)

When the four selections have been made, user can click "Start" button, and model export is started. A report of the export is written on right hand side of the window in real time, and user may cancel the export at any time. When export is finished, a note of it is written in the end of the report. If "export and open" alternative was selected above, Vulcan will now open.

Depending on the size of the analysis model, export may take from one second to a few minutes. The analysis model of Tekla Structres may have some features Vulcan cannot support. "Rigid links", for example, are an unfamiliar concept for Vulcan. In these cases, the unsupported features will be ignored in the export, and the report written on the right hand side of VulcanExport window will contain "warnings". Analysis model can still be opened in Vulcan, but user should very carefully check all warnings before running Vulcan analysis. If report contains "errors", then the export has totally failed and the analysis model cannot even be opened in Vulcan.

4.1.2 Implementation

Class diagram in figure 4.4 illustrates the structure of VulcanExport program. Class 'Program' is simply a dummy class to provide a static Main() function, that is called when program is started and then creates an instance of 'VulcanExport' as a Windows application. Class 'VulcanExport' has all user interface functionality in it. 'VulcanExport' has a member object 'm_oExportAnalysisModel', which is an instance of class 'ExportAnalysisModel'. When user has clicked 'Start' button, ExportAnalysisModel's 'ReadAndExport()' method is called and program starts doing what it is meant to do.

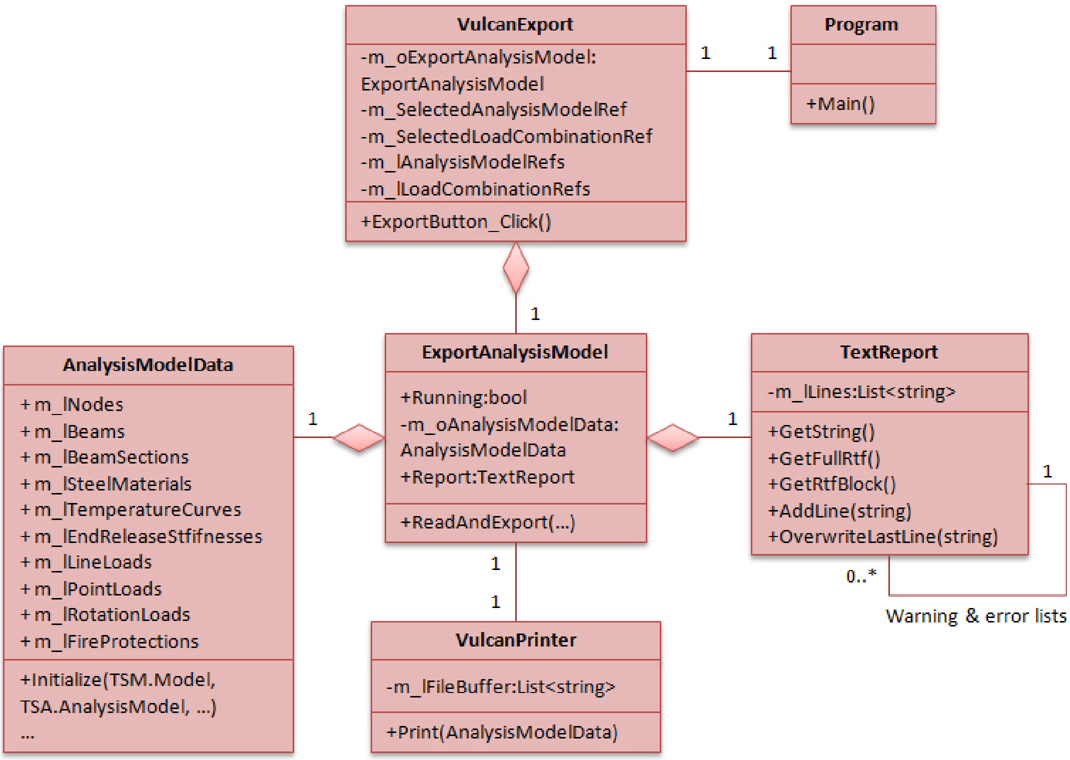

Figure 4.4: Simplified class diagram of VulcanExport macro.

'ExportAnalysisModel' has a member object of 'AnalysisModelData' class, that stores all the data read from Tekla Structures analysis model. 'AnalysisModelData' also implements all the functionality to read it's data from Tekla in it's public function 'Initialize()', that calls several private functions not presented in figure 4.4. When all data is read in, 'ExportAnalysisModel' creates an instance of 'Vulcan-Printer' class that holds all functionality to write Vulcan format text files. 'Vulcan-Printer' has one public function, 'Print()', that gets the whole 'AnalysisModelData' object as parameter and then calls several private functions not presented in figure 4.4 to write a text file first in it's own buffer and then save it in .vul format.

The report of the export, seen on right hand side in figure 4.3, is written during initialising 'AnalysisModelData' and the Vulcan file printing in a public member object of 'ExportAnalysisModel', 'Report'. 'Report' is an instance of class 'TextReport'. Error and warning lists within the report on user interface are temporarily saved in separate instances of 'TextReport' class for practical reasons, so for a moment they are "reports within a report".

In order of the user interface not to freeze while export is being executed, and for the report to be updated in real time, VulcanExport macro runs in three threads. There is one thread for the user interface functions (for example user clicking 'Cancel' before the export is finished', one for the actual work (ExportAnalysisModel.ReadAndExport()), and one for the updating of the report text box.

4.2 Modifications to FDSadapter macro

4.2.1 Usage and functionality

In order to be able to use Vulcan in NFD environment, the existing FDS adapter -macro needed to be modified to support Vulcan too. Figure 4.5 illustrates the user interface after modifications. Vulcan file (.vul) may now be selected as a source of temperature points. New selection for method of simplification of temperature points is also added.

User must select these two selections, the source of temperature points, and the simplification method. Also, user needs to select simulation time. Simulation time $0s$ may be used for previewing the model in Smokeview program before the time-consuming FDS analysis. If analysis has already been run, but Vulcan or Scia file needs updated temperature curves for some reason, user may select 'FDS ready, only XML/.vul output' to get the results without running FDS analysis again.

When user clicks 'Start' button, FDS input file is written, FDS started, and after the FDS-analysis is finished, the results are viewed in Smokeview program.

4.2.2 Implementation

FDS adapter has not been programmed in object oriented way, so class diagram is not a good way to illustrate it's structure. Instead a simplified sequence diagram in figure 4.6 shows the interactions between Tekla, FDS adapter, FDS, Smokeview, and Scia/Vulcan during FDS adapter's execution. Sequence diagram can be read so, that exceqution starts from top left cornet of the diagram. Each vertical line represents a 'lifeline' of a program or file, with the name of program or file on

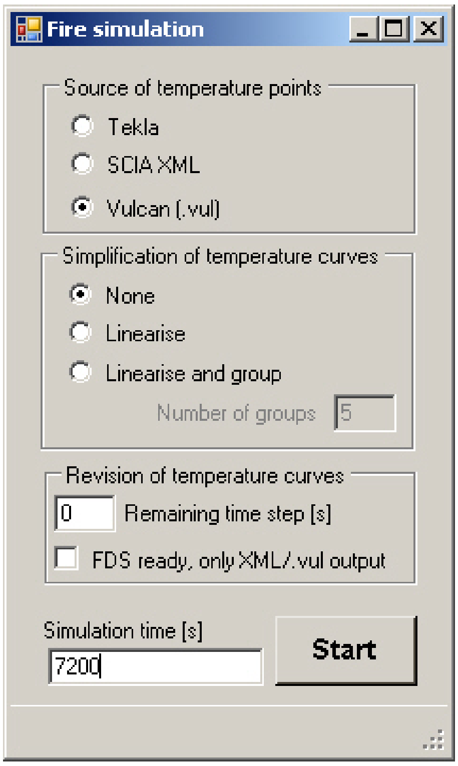

Figure 4.5: FDSadapter dialog.

top of the line. Vertical boxes are programs being executed, and horizontal arrows interactions between them. Time flows from top to bottom, so that interactions lower in the diagram take place after the interactions above them.

FDS adapter can be started from Tekla Structures in a very similar way as Vulcan Export, see figure 4.2. Once user has set the variables and clicks 'Start', FDS adapter creates an input file for FDS and starts writing in it. Input file is a plain text file containing FDS keywords, see [29]. Data is read from Tekla model (selected parts and grids and their user defined attributes). Transferring grids and parts read from Tekla into FDS input format is naturally a more complex task than the two arrows in figure 4.6 show. That is not in the focus of this study.

Beam mid co-ordinates are read from either Vulcan file (.vul) or XML-file exported from Scia, and these points are written in FDS input file as temperature output points using 'DEVC' keyword, see interaction 'Read beam mid-point' in 4.6. When FDS is run, in writes temperatures of these output points in a spreadsheet file (.csv) at different times of the analysis. When analysis is finished, FDS adapter then reads this file (interaction 'Read temperature curves' in figure 4.6), processes the data and writes it in either Vulcan files or XML files that can then be imported in Scia (interaction 'Write temperature curves' in figure 4.6).

Finally, FDS adapter opens Smokeview program to visualise the results of FDS analysis. When user closes the Smokeview, also FDS adapter will close itself.

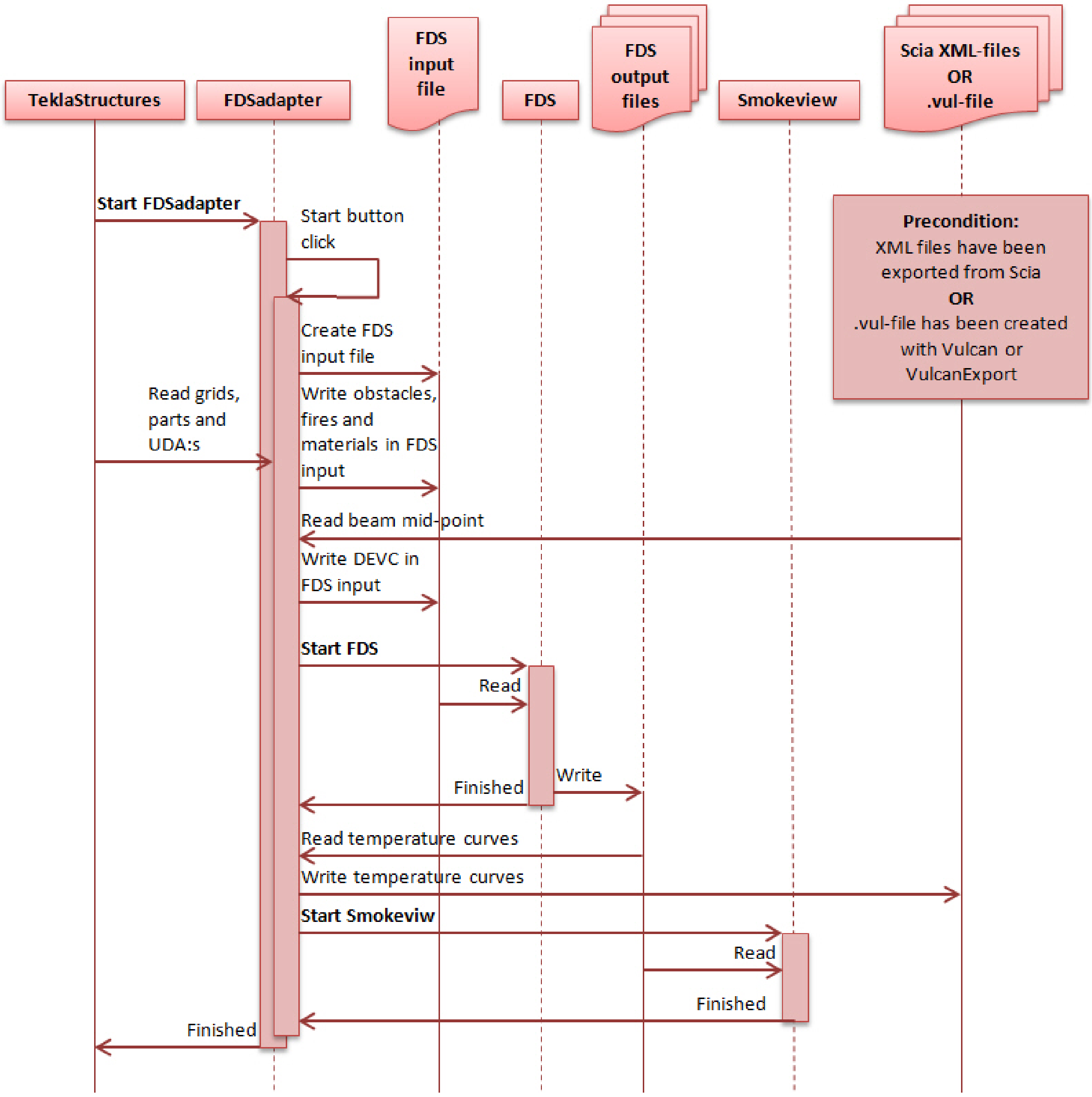

Figure 4.6: Sequence diagram of interactions between FDSadapter and other programs.

In this study, reading beam mid-points and writing temperature curves needed to be re-programmed, as they were previously done only with Scia XML-files. Also the processing of temperature data in FDS adapter between interactions 'Read temperature curves' and 'Write temperature curves' was re-programmed to add functionality to linearise and group temperature curves with algorithms presented in section 5.1.

5. TECHNICAL DETAILS ON INTEGRATING VULCAN IN THE NFD ENVIRONMENT

Chapter 4 described the macros that were programmed to integrate Vulcan into NFD environment - how to use them and how they were implemented. This chapter concentrates on some of the algorithms, limitations and technical details embedded in these macros.

5.1 Linearisation and Grouping algorithms

One goal of this study was to develop a method of simplifying the temperature data outputted from FDS. Standard temperature output from FDS is a temperature curve with one thousand time-temperature points. The curve has a lot of noise in it, and the amount of temperature data per curve, as well as using different temperature curve for each finite element in the analysis model increases calculation time of structural analysis, as described in [5]. For more human reasons, engineers and authorities also often find simplified temperature curves easier to work with, than the detailed curves with a lot of noise.

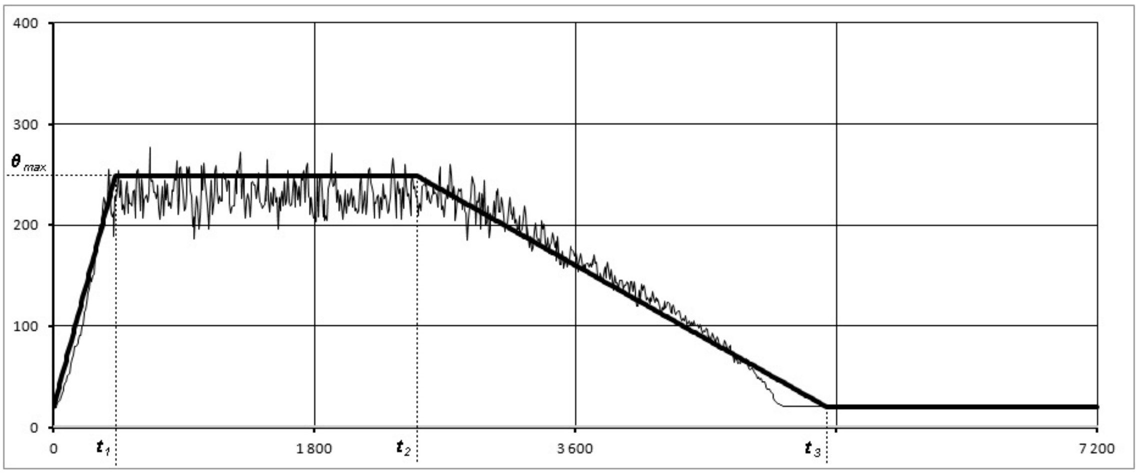

Figure 5.1: An FDS output time-temperature curve and a linearised version of it. Each linearised curve can be presented with four parameters: θ_{max}, t_1, t_2 and t_3.

Two simplification methods were implemented in FDSadapter -macro. First, a method to linearise each temperature curve with four lines as in figure 5.1. Second, a method to group multiple temperature curves to be presented by one simplified curve, so that several finite elements in a structural analysis model can have the same simplified temperature curve.

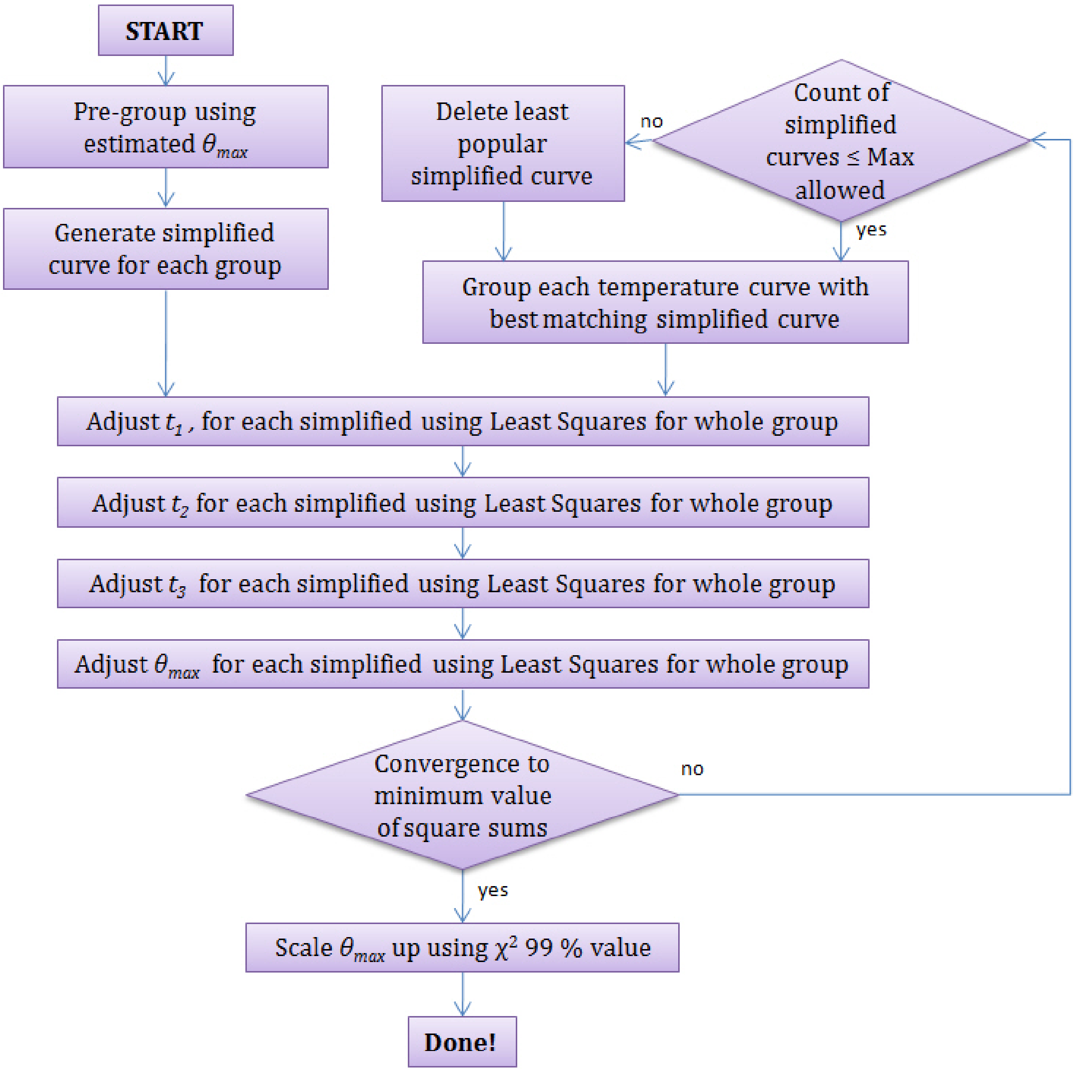

Figure 5.2: Algorithm for grouping and linearising temperature curves

Flowchart in figure 5.2 illustrates the algorithm for grouping. Adjusting the parameters of a simplified curve to best fit the group of temperature curves represented by it is based on Least Squares Method. A Square sum (5.1) is calculated for the simplified curve, and the parameters are adjusted so that the square sum is minimised. Because the mathematical representation of simplified curve is not differentiable, minimum value cannot be found using derivative of the square sum function. Instead, golden section search is used separately for each of the four parameters in order to find the parameter value that gives minimum square sum. Separate search for each parameter works well, because the square sum function for this problem by nature has one very clear minimum instead of multiple peak values. For a more complex problem, more advanced algorithms would be needed in order to find the minimum of square sums.

$$\sum_{j=curve} \sum_{i=timestep} (\theta_{simplified}(t_i) - \theta_j(t_i))^2, \tag{5.1}$$

For the square sum minimisation to work, it is essential to find a reasonable initial estimate for the simplified curve maximum temperature θ_{max}. This is done by an algorithm that tries temperatures with 50 Celsius degree steps from 2000 degrees down to 20 (last step only 30 degrees). The highest temperature, for which the temperature curve is above the tried temperature is more than 30 % of the time the temperature curve is above 50 degrees, is selected as the initial θ_{max} for that temperature curve. An example is presented in figure 5.3.

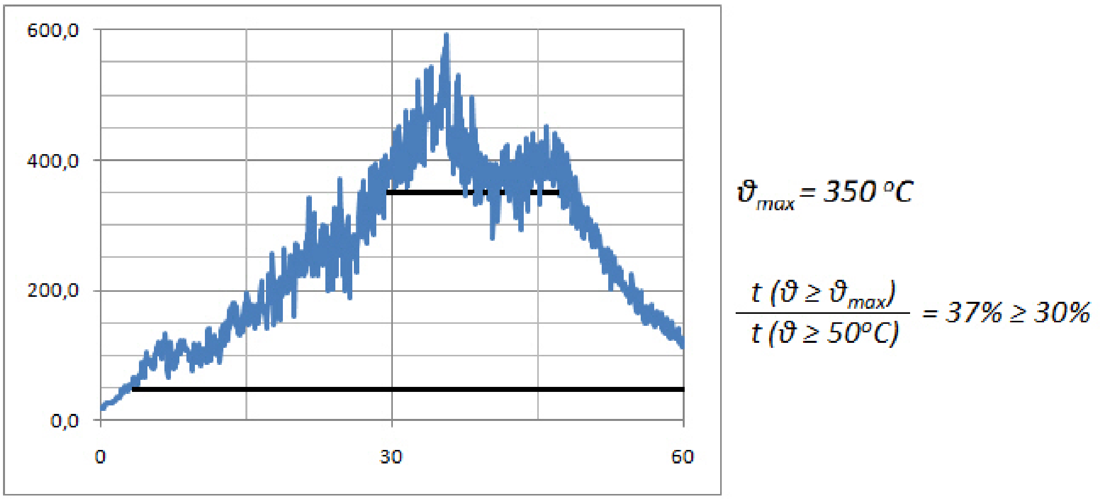

Figure 5.3: 350 is the highest θ_{max}-candidate, for which temperature curve is above θ_{max} more than 30 % of the time, when temperature curve is above 50 Celsius degrees. 350 is selected to be used as the initial estimate of θ_{max}.

All temperature curves with the same initial θ_{max} are grouped in the same group for the first iteration round. If number of groups is larger than the maximum number given by the engineer on FDSadapter user interface, then one simplified curve is deleted on each iteration round until the number is sufficient. On each iteration round, the temperature curves are re-assigned to best matching simplified curves so that curves form the deleted group can find a new group and other curves are allowed switch groups during iteration.

When number of groups is sufficient, no temperature curve has switched groups on latest iteration round, and no significant decrease of square sums took place on last iteration round, then the algorithm has converged to a solution. At this point, maximum temperatures of the simplified curves are mean values of the group. In engineering projects, maximum values are needed. θ_{max} of each group is scaled up using 99% value of group's χ^2-distribution. Figure 5.4 shows a temperature curve

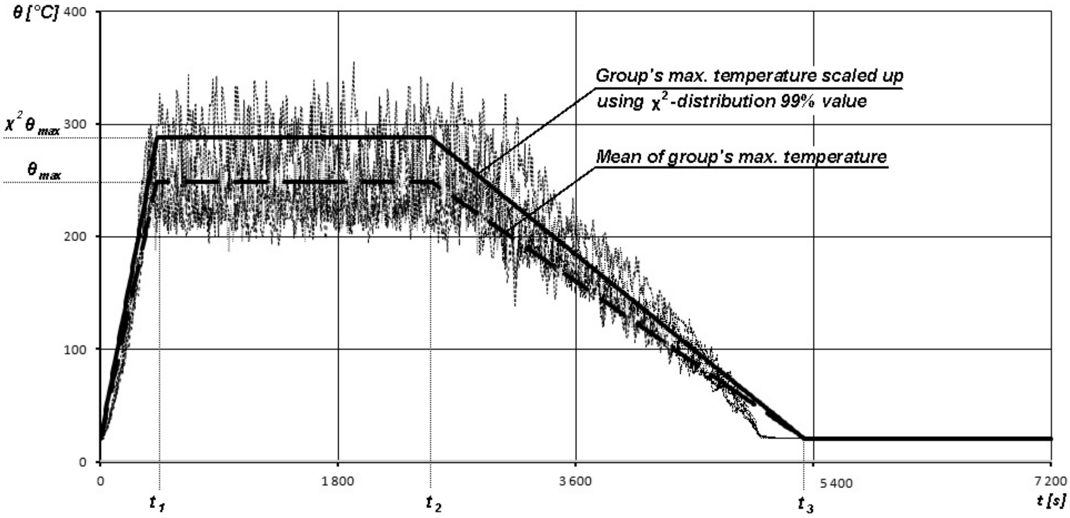

Figure 5.4: A temperature curve group with it's simplified curve with with mean θ_{max} and the final simplified curve with θ_{max} scaled up.

group, it's simplified curve with mean θ_{max} and the final simplified curve with θ_{max} scaled up.

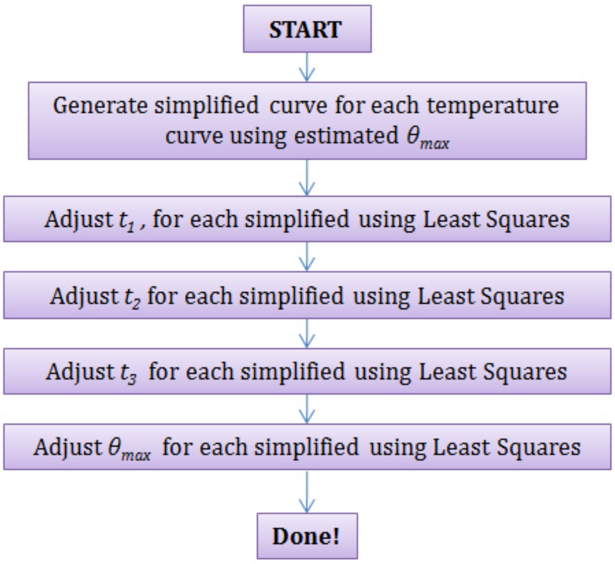

Figure 5.5: Algorithm for only linearising temperature curves.

Flowchart in figure 5.5 illustrates the algorithm for linearising temperature curves without grouping. θ_{max}, t_1, t_2 and t_3 are adjusted with same method as in grouping algorithm, but this time "group" only has one temperature curve in it. Since no checks of a temperature curve matching a specific group are needed, only one iteration round is needed for convergence. Because each simplified curve represents only one temperature curve, there is no data to use χ^2-distribution to scale up θ_{max}. For only one temperature curve, best simplification naturally is the mean value and χ^2-scaling is not needed.

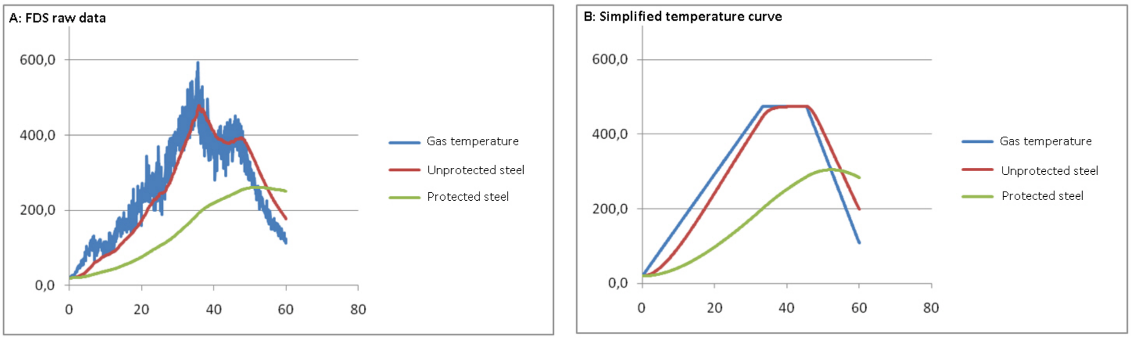

Figure 5.6: Steel temperatures calculated using FDS raw data and simplified temperature curve.

Figure 5.6 shows steel temperatures of same steel section calculated using FDS output data and simplified curve of the same data. It can be seen, that for protected steel simplification does not affect shape or magnitude of the steel temperature curve. Unprotected steel section is more sensitive to small variation and noise in gas temperature. Highest temperature and curve shape during the first minutes of fire match well, but the overall shape of the unprotected steel temperature curve based on simplified gas temperature differs from the one based on more detailed gas temperature data. Simplified model can be used to verify that an unprotected steel member is clearly fire resistant enough - or is clearly not fire resistant at all. If the level of fire resistance is not clear, i.e. utilisation ratio of the member is close to $1,0$, simplification may be too rough and it's use should be carefully considered.

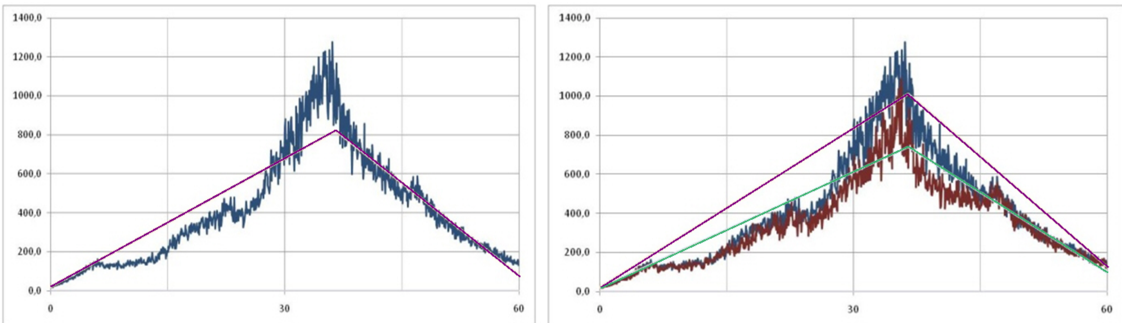

Figure 5.7: Simplified temperature curve of a temperature curve that is not easy to linearise.

Figure 5.7 shows a temperature curve that has a concave shape before the peak. On left hand side the simplified curve peak is considerably lower than the peak value of the original curve. This is because the least squares method by it's nature fits the linear shape to represent the mean value of the curved temperature function. Result is not satisfactory, because usually the peak value is most interesting in design, and in this case the greatest error is at the peak. On right hand side simplified curve is first fitted to a temperature curve group of two curves (green line) and then scaled upwards using χ^2-distribution (purple line). Now error of peak value reduces significantly, because scaling factor is large due to high variance in simplified curve compared to original data.

The use of χ^2-distribution 99 % values in grouping algorithm generally reduces the ucertainty of simplified model by scaling results to "safe" side. The worse the simplified model fits the original data, the larger sacling factor is used. χ^2-distribution is more reliable the more members are represented by one simplified curve. If grouping algorithm groups only one member to be presented by one simplified curve scaling factor is 1.0, so mean values are used. This should be noted by designer, if temperature data is known to be hard to simplifiy, such as the concave curve in figure 5.7.

A conclusion may be drawn, that for protected steel structures simplified gas temperature curves may be used with few or no restrictions, but for unprotected steel the error caused by simplification should be carefully considered, especially if utilisation ratios are close to $1,0$. Also, if only one member is grouped to be represented by one simplified curve, then it is advisable to check how well this simplification actually represents the original data.

5.2 Profile mapping in VulcanExport

5.2.1 Cross section identification

A common dilemma in transferring a beam model (geometry model or analysis model) between two design programs is the different representation of cross sections in different programs. Most modelling and analysis programs define beams with a simple line from start point to end point, with a reference to a cross section in some kind of cross section database. An intuitive way to transfer the beam to another program would be to export the start point and end point co-ordinates and the cross section name. Problems occur, when it cannot be known what kind of naming system the recieving program uses. Tekla profile catalog, for example, uses name "HEA200" for one standard profile, but Vulcan section database uses name "HE200A" for the same profile.

Attempts to standardise naming of profiles and steel grades have been made, SteelBase project by FSCA just to mention one [30]. Still, variety of naming systems between programs remains.

If the profile databases in Tekla and in Vulcan were static, a mapping algorithm for cross section data transfer would be easy to implement. In most modelling and analysis programs, however, user can modify the cross section database by adding and editing cross sections. Therefore, making assumptions of the profile names is not a good solution.

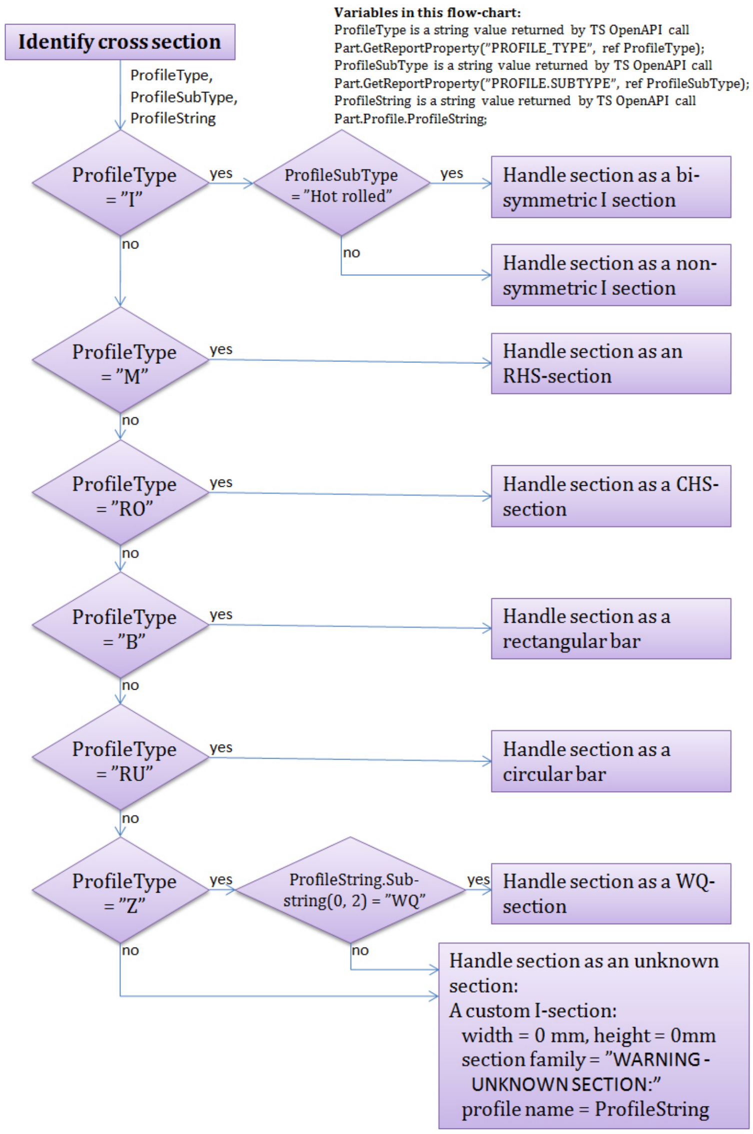

Figure 5.8: Flowchart of the cross section identification algorithm in VulcanExport.

A standard solution to this would be writing a specific clear language mapping file definig the cross section names in both Tekla and Vulcan. VulcanExport macro would then check this file when converting the cross section names from Tekla format to Vulcan format. If conversion wouldn't work with some specific cross sections, user could freely edit the mapping file.

VulcanExport macro uses a different approach. Because Vulcan input file accepts "custom" cross sections that can be given all cross section attributes, not only the name, the attributes can be read directly from Tekla. Same transfer method can be used for both standard and custom cross sections. If a standard cross section is stored in Vulcan database, cross section will be handled as a standard section. If not, cross section will be handled as a custom cross section. This way Vulcan end of the data transfer link is very generic.

Tekla end of the data transfer is slightly more complicated. Standard way of reading cross section properties from the model by TeklaOpenAPI is using the GetReportProperty()-function. Problem with this is, that different cross section types have different "report properties" that need to be read, and identifying the type of a specific cross section is not always simple. For example, the top flange width of a bi-symmetric I-section is read with TeklaOpenAPI command:

```
double dUserVariable = 0.0;
Part.GetReportProperty("PROFILE.WIDTH", ref dUserVariable);
```

but if I-section is not bi-symmetric different report property must be used:

```
double dUserVariable = 0.0;
Part.GetReportProperty("PROFILE.FLANGE_WIDTH_1", ref dUserVariable);
```

All I-sections have the same report property `PROFILE_TYPE`, so another report property `PROFILE.SUBTYPE` needs to be used to identify if profile is bi-symmetric or not.

Tekla supports a large variety of cross sections, and user can define new cross sections without limitations. It is impossible to program a data transfer link that would support all imaginable sections, so cross sections supported by VulcanExport were limited to bi-symmetric and unsymmetric I:s, RHS, CHS, rectangular bars, circular bars and WQ-sections. All other sections are handled as "unknown sections" and given properties of a custom I-section with 0 mm heght and width. This way beam co-ordinates will transfer to Vulcan correctly, but user won't be able to run analysis without first setting the unknown sections some attributes manually.

Cross section identification algorithm is probably the part of VulcanExport most vulnerable to changes in future versions of Tekla Structures, or new cross sections defined by Tekla user. Because of this, it has been carefully documented in figure 5.8

5.2.2 Representing non-standard cross sections in Vulcan file format

Representation of a cross section in Vulcan is dependent on section type. There are three types, I Section, ASB Section and Non-Standard. The two first types are further divided into section families, such as UB sections and UC sections, but from programming point of view these make no difference -each section in UB family is just a "custom" I section with pre-defined family name and size name saved in section database. New families may be added by adding new section database files on the computer, and sizes of a family may be changed by editing the content of these files with a regular text editor. Figure 5.9 illustrates the division of sections by type, family and size in Vulcan.

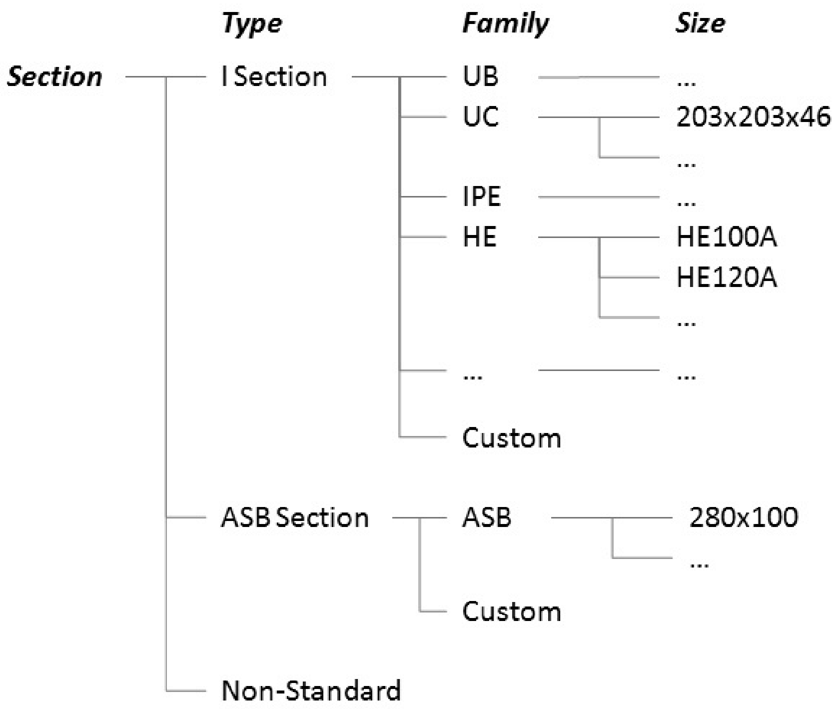

Figure 5.9: Cross sections supported by Vulcan. Each family has several sizes in it, and only a few samples are shown here.

Section family and size are irrelevant from programming point of view, since they don't affect the format in which sections are saved in .vul-files. The saving format is dependent on section type. Above, figure 5.8 showed how Tekla cross sections were identified as different types to be handled in different ways. Bi-symmetric I sections can be saved as Vulcan I Section type and non-symmetric I sections as Vulcan ASB-section type. All other supported Tekla sections are saved as Vulcan non-standard type. The Tekla sections that are not supported are saved as Vulcan I sections, but with 0 dimensions so that user cannot analyse the model before replacing the non-supported sections manually.

Saving cross sections as Vulcan I sections or ASB sections is trivial, but saving as non-standard section requires a bit more attention. Listing 5.1 shows part of a .vul-file that specifys two cross sections for the model in case. In the beginning of the listing there are nine comment lines explaining the meaning attributes in natural language, followed by the actual code. The code begins with number two meaning there are two different sections. Then there are 25 lines specifiying a WQ beam as a non-standard section. After that, there are four lines specifying an IPE-beam as an I section.

Listing 5.1: Vulcan file (.vul) presentation of two sections, a WQ320-6-30x240-20x510 specified as non-standard section, and an IPE300 specified as a standard I section. Some decimals and white spaces have been removed form listing to better fit the page.

```
//  First line is Number of Steel Sections
//  Then Number, WidthDivisions, DepthDivisions, Depth, TopWidth,
//    BottomWidth, TopFlangeThickness, BottomFlangeThickness,
//    WebThickness, SectionFactor, Type
//  Then WidthSizes
//  Then DepthSizes
//  Then (if section is in Undefined Family) Material Matrix
//    (0=Air, 1=Steel, 2=Concrete, 3=Rebar)
//  Then (if section is in Undefined Family) Property Matrix
<BEAM SECTIONS>
    2

    1 10 10 340.00 510.00 510.00 30.00 20.00 6.00 0.18 0
 WQ-Beam
 WQ32-240
   64.50 64.50 6.00 60.00 60.00 60.00 60.00  6.00 64.50 64.50
   30.00 36.25 36.20 36.20 36.25 36.25 36.25 36.25 36.25 20.00
    0     0     1     1     1     1     1     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    1     1     1     1     1     1     1     1     1     1
    0     0     1     1     1     1     1     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    0     0     1     0     0     0     0     1     0     0
    1     1     1     1     1     1     1     1     1     1

    2 10 10 300.00 150.00 150.00 10.70 10.70  7.10 0.229 1
    IPE
    IPE300
    1
</BEAM SECTIONS>
```

I section geometry is hard coded in Vulcan, so just one line is needed to give all the geometric properties of the section, such as height, width, flange thickness and web thickness. Two other lines specify section's family and size name, and last

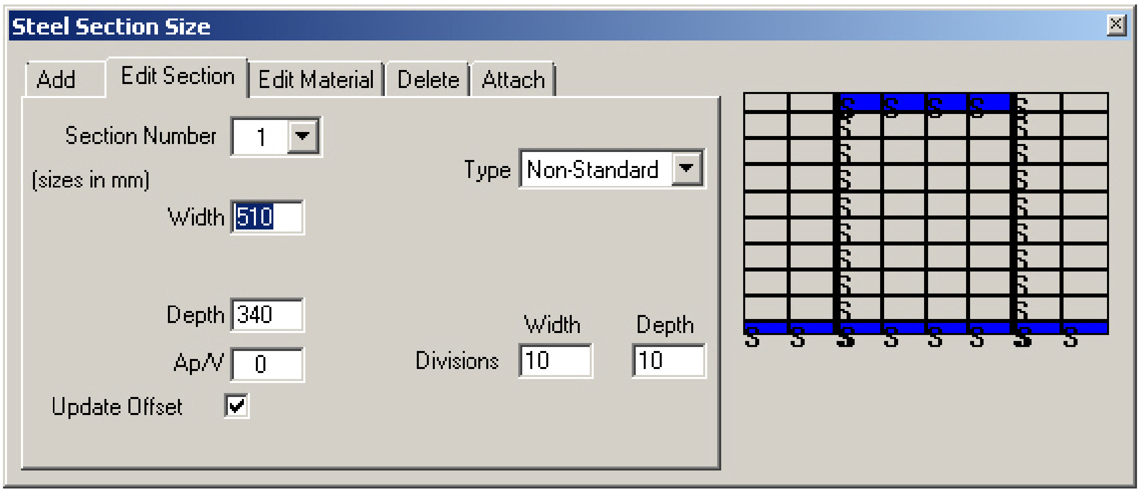

Figure 5.10: WQ320-6-30x240-20x510 cross section on Vulcan user interface.

line material. Non-standard section geometry is not hard coded, so special material matrix and property matrix need to be written in .vul-file. In listing 5.1 WQ-beam geometry is specified in a 10x10 material matrix, but also other matrix sizes are possible. Matrix itself has values 0 for air and 1 for steel. First line of matrix has six 1:s to represent top flange of WQ-beam, and lines from second to ninth have only two 1:s to represent the two webs of the beam. Tenth line has ten 1:s to represent the wide bottom flange of the WQ beam. Figure 5.10 shows this cross section on Vulcan user interface.

In listing 5.1 material matrix is followed by a similar looking property matrix so, that the two matrices actually look as one. Property matrix begins after the tenth line, that is all 1:s. If the model had two specified steel properties, S235 and S355 for example, then the property matrix might have number 2:s referring to the second steel material on the material list, but in this case all steel refers to 1 and the property matrix is all 0:s and 1:s, and identical to material matrix.

The lines before the matrices in listing 5.1 refer to the sizes of each "cell" in the material matrix. First two cells from the left are 64.5 mm wide, giving the bottom web extension total width of 129 mm. Third cell is only 6 mm wide, because third cell represents the 6 mm web of the WQ beam in case.

Understanding how material matrix and property matrix can be used, importing rectangular bars, rectangular hollow sections and WQ sections to Vulcan is straightforward. In VulcanExport macro 10x10 matrices were used, and the cell sizes were modified to have the plate thickness correctly. Rounded corners of the rectangular hollow sections cannot be modelled in this way. Even standard I profiles in Vulcan are missing the effect of rounded corners, so this was not considered a problem compared to the current accuracy of Vulcan analysis.

Importing circular bars and circular hollow sections as Vulcan non-standard sec-

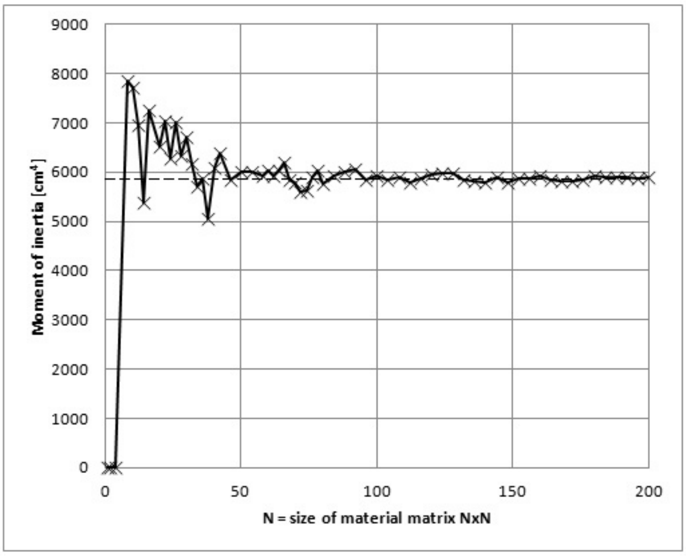

Figure 5.11: Moments of inertia of CHS 273x8 section represented by different sized NxN material matrizes. Accurate moment of inertia $5852cm^4$ displayed by dashed line.

tions requires a bit more thinking. Cells of the material matrix are squares, and representing a round form with squares is a bit problematic. Several different matrix sizes from 10x10 to 200x200 were tried and cross section areas and moments of inertia were calculated for the "squared estimates" of a few circular hollow sections. Figure 5.11 shows the (I, N) relations of a CHS 273x8 section, where N is the size of a NxN -matrix. The tested cases gave confidence, that with matrix size 100x100 or larger all error should be within satisfactory 5% limits.

Listing 5.2: A 22x22 material matrix representing a CHS 273x8 section. 1 represents steel, 0 represents air. Calculated moment of inertia for this simplified section is $7031cm^4$, while accurate value is $5852cm^4$. With matrix size 100x100 accuracy may be considered good enough.

0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Material matrix size 100x100 was selected to be used in Vulcan with all circular sections. Large matrix has the disadvantage of making editing of .vul files with a regular text editor somewhat harder. On the other hand, as Tekla and VulcanExport macro can be used to produce the Vulcan models, need for manual editing of the files should reduce significantly.

5.3 Connection design in fire using 3D component method

One of the goals for this study besides integrating Vulcan into Natural Fire Design environment, was taking a step towards connection design in fire using 3D component method. Component method is used for determining rotational stiffness (and rotational strength) of semi-rigid connections. Method has been presented for planar cases in [9] and [10], and extended to 3D in [11], [12] and [13].

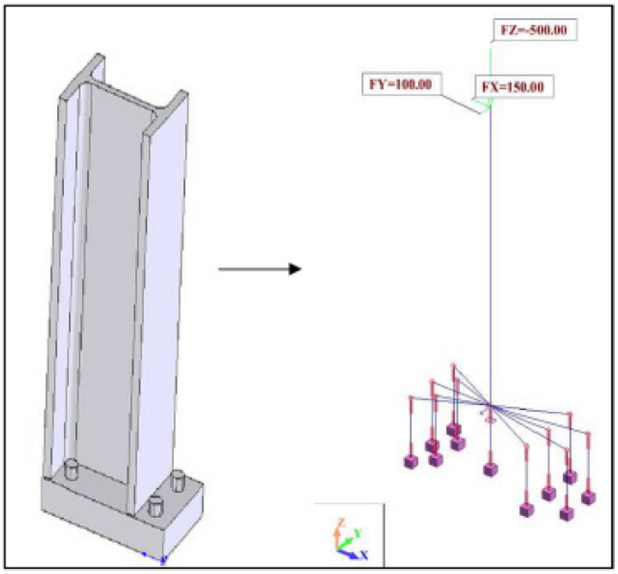

Figure 5.12: 3D component method representation of a column base plate connection [12].

Eurocode 3 part 1-8 [10] allows modelling of steel connections as a hinge, semi-rigid connection or rigid connection. If connection is semi-rigid, the connection stiffness affects the stress distribution in the whole structure. If rigid or semi-rigid connections are used in bracing of the whole building, stiffness may affect the building being sway or non-sway. For connections of compressed members, such as column base connections, connection stiffness affects the buckling length of the member. In case of fire, semi-rigid connection stiffness is temperature-dependent and the temperature of the connection may thus affect greatly the resistance of whole structure.

In [12] a 3D component method approach was developed for column base plate connection. Component model is built so, that the compressive behaviour of concrete below the compressed flange is modelled with compression only truss finite elements in a finite element analysis program capable of non-linear analysis. Tensile behaviour of the anchor bolts and the bending of the base plate connecting anchor bolts to column are modelled with tension only truss finite elements. Tension only and compression only members were connected to column base with rigid links, and the other end of the truss was supported as a boundary condition, see figure 5.12. Cross section area and elastic modulus of the elements are adjusted so, that the elements match the stiffness factors or spring constants specified by rules in EN 1993-1-8 [10]. As the rules are originally for planar case, some modifications are needed for 3D case, for example analysing individual bolts instead of bolt groups.

The 3D component column base connection was then programmed in Tekla as an Open API -connection in [13]. Connection could be modelled in Tekla and then exported to an analysis program together with the rest of the model. Connection macro was called Anchor Bolt Connection, and the capacity check macro 7DResults.

In this study, one of the goals was to integrate also Anchor Bolt Connection macro into Natural Fire Design environment, so that the semi-rigid column bases could be analysed in fire conditions with Vulcan. Unfortunately Vulcan does not support tension only or compression only elements, and the goal could not be achieved.

The matter was discussed with Roger Plank from Vulcan Solutions Ltd. and University of Sheffield and it was revealed, that some component method based connections will be included in future versions of Vulcan [31]. This approach is likely to be more user friendly than using 'rake models' consisting of rigid links and compression / tension only trusses. It only postpones into future any results desired.

5.4 The possibility of integrating research version of Vulcan

The research version of Vulcan is constantly being developed, and it has features that commercial Vulcan does not have. One of the most interesting of these features at the time of this study was the static-dynamic analysis allowing Vulcan to get past the first loss of stability and perhaps re-stabilise a short dynamic phase of partial collapse. Another were the new connection models being developed.

An opportunity occurred to do part of this thesis in University of Sheffield with access to the research version of Vulcan and teaching from it's developers. Purpose of the visit was to investigate, whether the research Vulcan also could be integrated with BIM. The case studies revealed, that as research Vulcan is less user friendly than the commercial version, and the static-dynamic analysis extremely time-consuming, so that generating the input files from large building information models would not be as useful as it is with the commercial Vulcan. A separate report of the Sheffield visit is available [32].

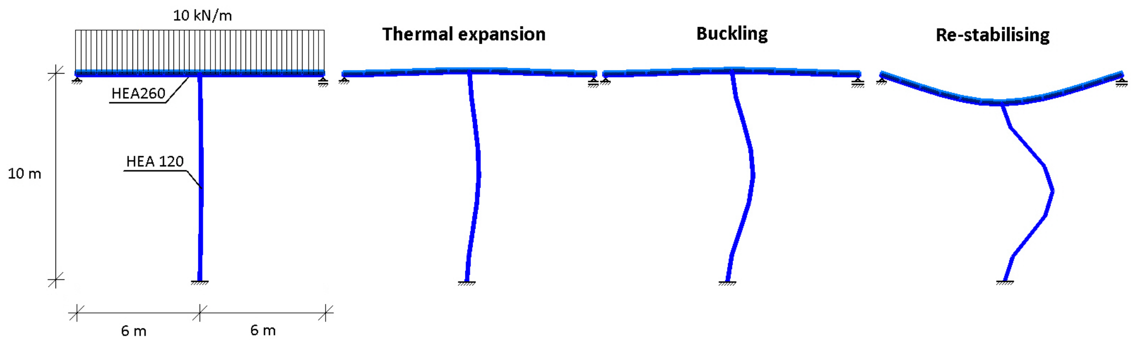

Figure 5.13: Example structure for static dynamic analysis [32]

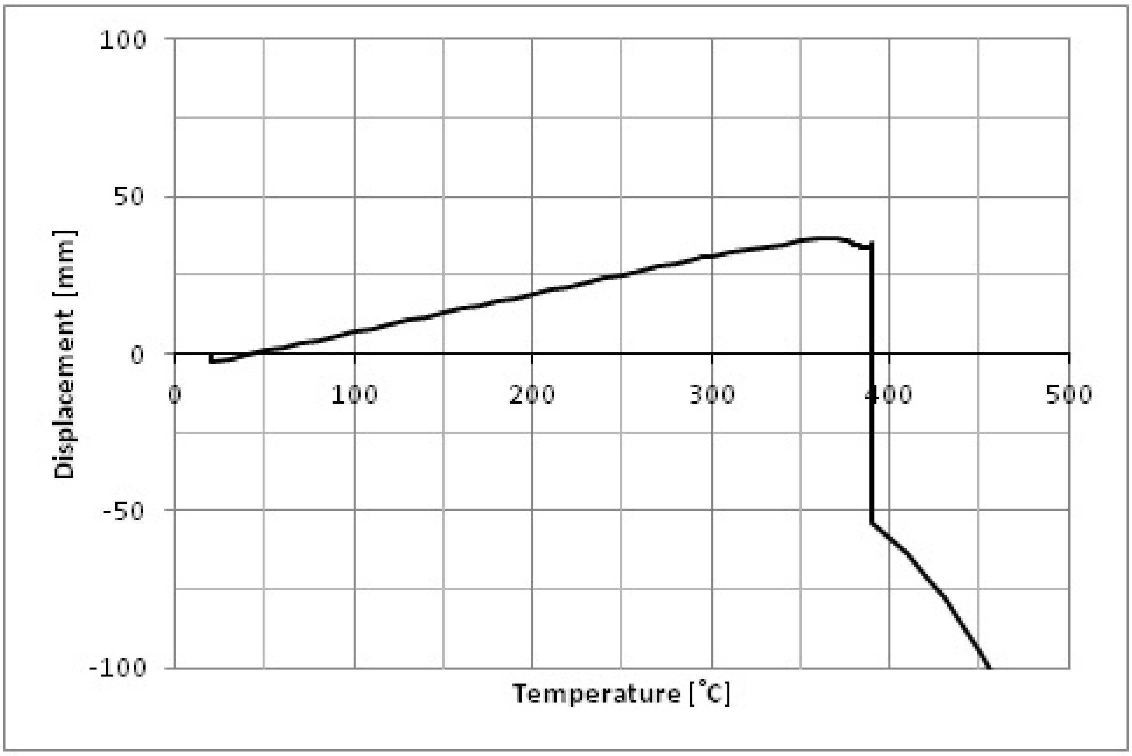

Figure 5.14: Vertical displacement of the beam mid node [32]

A simple example of the static-dynamic capability of the research Vulcan is presented in figure 5.13. It is is a continuous beam supported by a very slender column, which will buckle at an early stage of the fire when the beam still has plenty of bending capacity left. As the static analysis of commercial Vulcan would stop at the buckling of the column, dynamic solver of the research Vulcan can go past the sudden drop of the no longer supported part of the beam and then continue the static analysis of what now is a simply supported beam.

Figure 5.14 shows the vertical displacement of the beam mid node. The column buckling is clearly visible at 390^oC. After dropping rapidly about $90mm$, the beam can re-stabilise because of the bending capacity it has left.

6. CASE STUDIES

Two case studies are included in this thesis. First is a simple continuous beam, which serves to demonstrate the difference between linear and non-linear structural analysis. Second is a sports hall, that demonstrates the complete work flow of fire design of a building using Natural Fire Design environment.

6.1 Continuous beam

To illustrate the value of non-linear finite element analysis for statically non-determined structures in fire, an extremely simple case of a two-span continuous beam under uniform loading is studied (figure 6.1). Both spans of the beam are 4 meters long. All supports can rotate freely, all are fixed for vertical translation, and just one is fixed for horizontal translation. Uniform design load has a value of 20 kN/m, and a beam section CFRHS 160x80x6 is selected as it gives a realistic 0.8 utilisation ratio in normal temperature. Rectangular hollow section was selected, so that the effect of lateral torsional buckling could be ignored and simple plastic theory used.

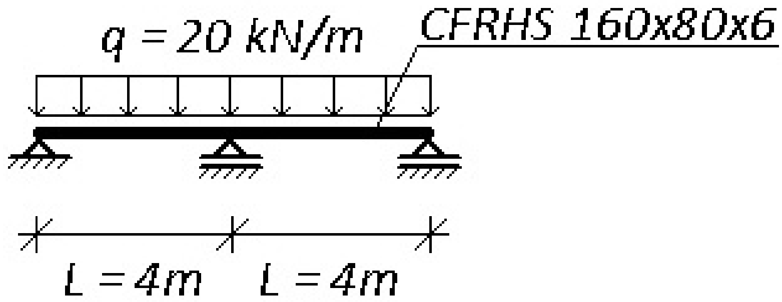

Figure 6.1: Continuous beam with uniform loading

Degree of statical non-determination for this structure is one, so, using plastic theory, it should form a mechanism and collapse as the second plastic hinge forms. Due to symmetric loading, second and third plastic hinge actually form at the same time. Linear finite element analysis, such as used in Scia, cannot take into account the re-distribution of internal forces when the first plastic hinge forms. Therefore resistance time of the beam in linear analysis is the time it takes for the first plastic hinge to form. With non-linear analysis, or in this simple case with hand calculations using plastic theory, it is possible to take the re-distribution into account and calculate the actual resistance time, the time it takes for the second plastic hinge to form. In following three sections the beam will be analysed first with hand calculations using plastic theory, then with Scia and finally with Vulcan using two different structural models.

6.1.1 Hand calculations using plastic theory

Figure 6.2A shows bending moment diagram of the beam at elastic state. As support moment reaches plastic bending capacity M_{pl}, first plastic hinge forms at support (see figure 6.2B). After this, support moment cannot rise any more, but span still has bending capacity left, and loading can increase. When also the span moment reaches plastic bending capacity, the second (and third) plastic hinge will form and beam will collapse (see figure 6.2C).

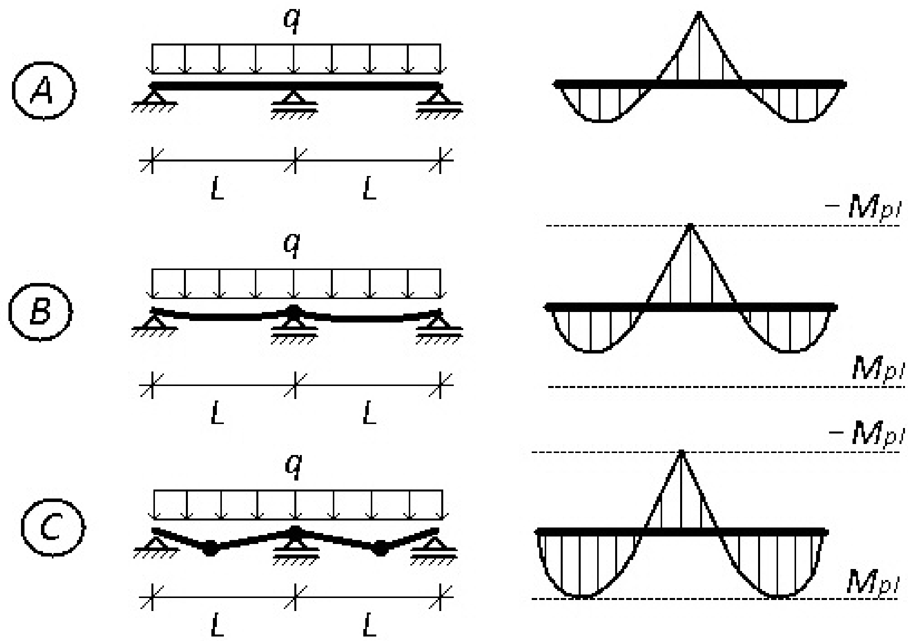

Figure 6.2: Moment diagrams at different states of loading

Typically in plastic analysis beam's M_{pl} is known, and analysis aims to find out the highest possible loading. In this case, loading is known, but M_{pl} will reduce as temperature rises, and the aim is to find out the highest steel temperature the beam can sustain. Plastic bending capacity of the beam is

$$M_{pl} = k_{y,\theta} f_y W_{pl}, \tag{6.1}$$

where $k_{y,\theta}$ is the yield strength reduction factor of EN1993-1-2 table 3.1 [7], f_y is the yield strength $355MPa$ and W_{pl} is the plastic bending resistance of the cross section. As stated in section 5.2.2, Vulcan cannot represent corner rounding of a rectangular hollow section, so a simplified value $W_{pl,Vulcan} = 139.6cm^3$ is used in Vulcan analysis. In Scia, however, it is very hard to use any simplified section instead of the ones on cross section catalog, so in Scia a more accurate value $W_{pl,Scia} = 132.3cm^3$ is used. In these hand calculations we use $W_{pl,Vulcan}$, because comparing results with Vulcan results will later be more interesting. M_{pl} depends on steel temperature, because $k_{y,\theta}$ is a function of temperature. M_{pl}-θ -relation of CFRHS 160x80x6 is shown as dotted lines in figure 6.4.

The temperature, at which first plastic hinge forms, can be determined by solving $k_{y,\theta}$ from equation 6.2

$$-0.125qL^2 = -k_{y,\theta} f_y W_{pl}. \tag{6.2}$$

Result is $k_{y,\theta} = 0.81$ and the equivalent temperature $\theta = 486^oC$. Support moment at this state is $M_{support} = -40kNm$ and span moment $M_{span} = +22.4kNm$.

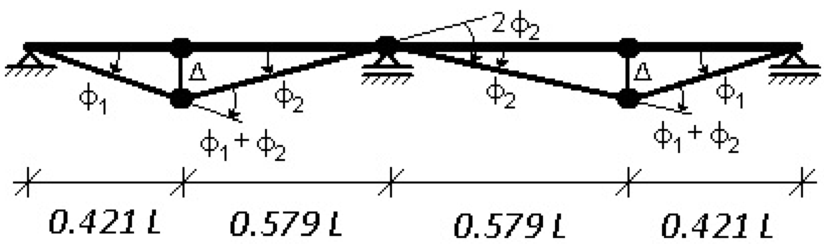

Figure 6.3: Collapse mechanism with virtual displacement Δ. Second plastic hinge is assumed in location $0.421L$ from beam end.

The temperature, at which the second plastic hinge forms, needs to be solved using virtual work principle and virtual displacement Δ in figure 6.3. To force virtual displacement Δ for plastic hinges on both spans, uniform load q must do external work

$$W_e = q\frac{0.421\Delta L}{2} + q\frac{0.579\Delta L}{2} + q\frac{0.579\Delta L}{2} + q\frac{0.421\Delta L}{2} = q\Delta L. \tag{6.3}$$

Simultaneously, moments in plastic hinges do negative internal work

$$W_i = -M_{pl}(\phi_1 + \phi_2) - M_{pl}2\phi_2 - M_{pl}(\phi_1 + \phi_2) = -11.659\frac{\Delta}{L}M_{pl}, \tag{6.4}$$

where

$$\phi_1 = \frac{\Delta}{0.421L} \tag{6.5}$$

$$\phi_2 = \frac{\Delta}{0.579L}. \tag{6.6}$$

Virtual work principle states, that

$$W_i + W_e = 0 \tag{6.7}$$

$$q\Delta L - 11.659\frac{\Delta}{L}M_{pl} = 0 \tag{6.8}$$

$$M_{pl} = \frac{qL^2}{11.659} = 27.447kNm \tag{6.9}$$

Beam has plastic bending capacity $27.447kNm$, when

$$M_{pl} = k_{y,\theta}f_yW_{pl} = 27.447kNm \tag{6.10}$$

$$k_{y,\theta} = \frac{M_{pl}}{f_yW_{pl}} = 0.55 \tag{6.11}$$

$$\theta = 574^oC \tag{6.12}$$

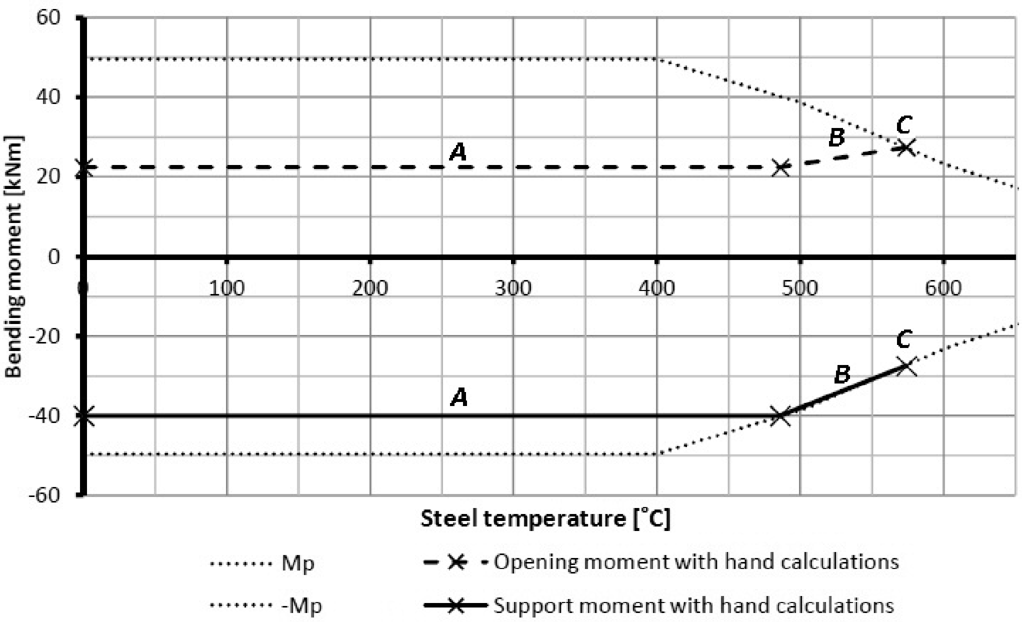

Figure 6.4: Beam moments at different temperatures: hand calculations using plastic theory. Letters A, B and C refer to the stages of figure 6.2.

The plastic hinge location at $0.421L$ in figure 6.3 is an estimate based on the location of bending moment peak value in elastic state, but is not totally accurate in plastic state and causes an insignificant error in results.

Figure 6.4 shows moments at support and spans as a function of time. Dotted lines show the plastic bending capacity of the beam as a function of time. It can be seen, that after the support reaches it's plastic capacity, moments can re-distribute in structure, and the structure can sustain another 88^oC rise in temperature before failing. Hand calculations were a rough simplification of the beam behaviour, and set an upper limit estimate of the beam capacity.

6.1.2 Linear analysis with Scia

Same structure was analysed in Scia for linear structural analysis in fire. Analysis gives bending moment distribution of figure 6.5. This distribution does not change as the temperature rises.

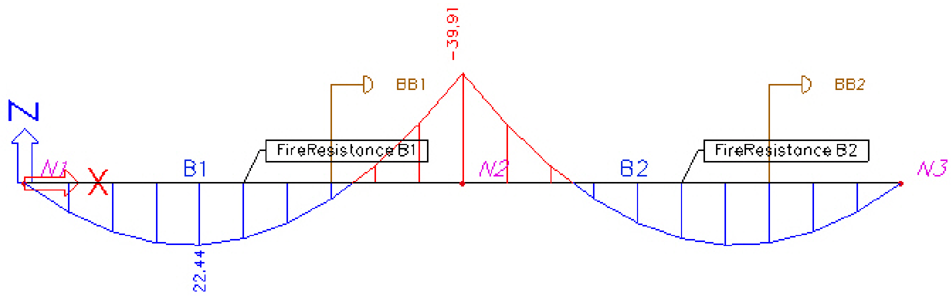

Figure 6.5: Beam moment diagram in Scia.

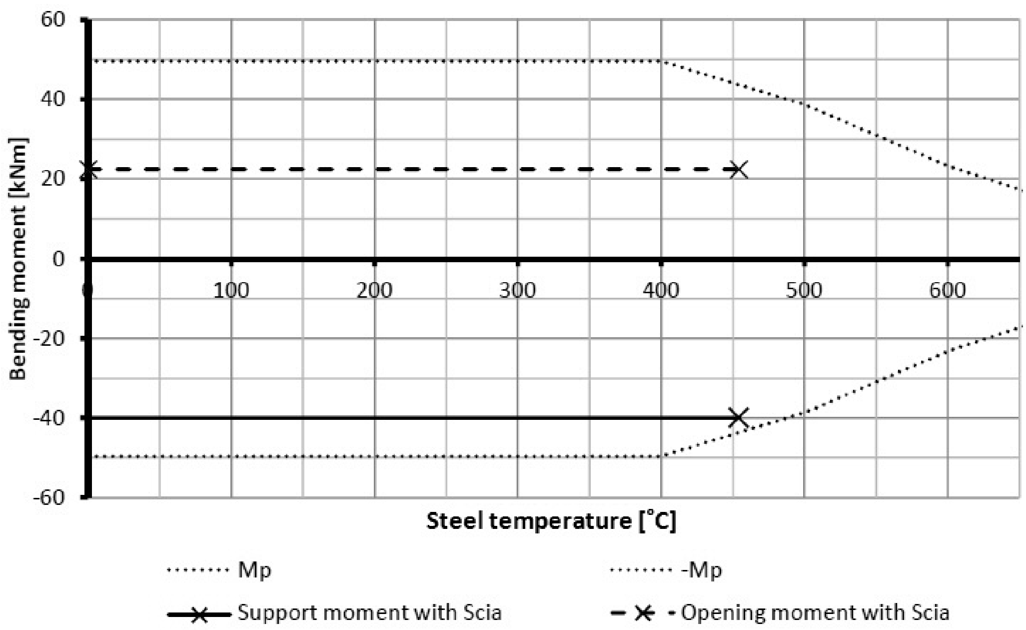

Figure 6.6: Beam moments at different temperatures: linear analysis using Scia.

Fire resistance check of the Scia version available for this study is based on preliminary Eurocode version ENV 1993-1-2 [33]. When stability check is not dominant, beam's design steel temperature is 454^oC. Critical check is combined bending and shear on support. Figure 6.6 shows the beam bending moments at spans and on support. Moments do not change as temperature rises, and fire resistance check will fail with any temperature higher than 454^oC.

6.1.3 Non-linear analysis with Vulcan

Same structure was analysed in Vulcan. Figure 6.7 shows the largest displacement at spans as a function of temperature.

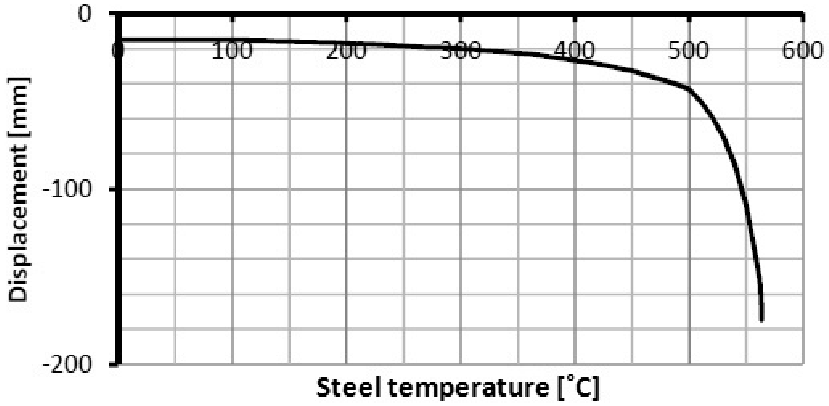

Figure 6.7: Displacement diagram in Vulcan analysis.

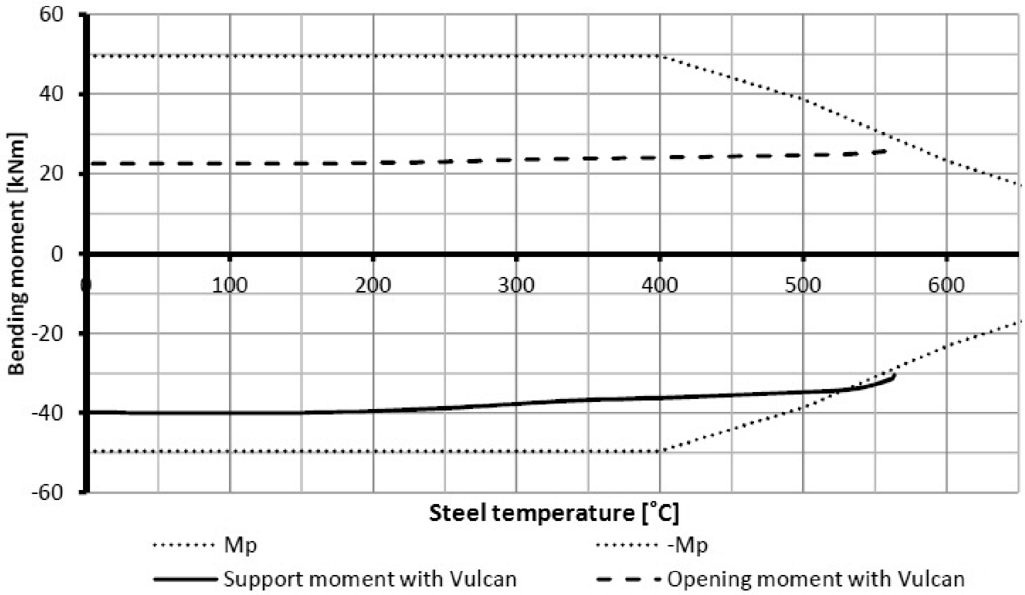

Figure 6.8: Beam moments at different temperatures: non-linear analysis using Vulcan.

Figure 6.8 shows the beam span and support moments as a function of time. It can be seen, that internal forces can re-distribute, and structure can sustain higher temperatures than in linear analysis of figure 6.6. Unlike in simplified calculations of 6.4, stress re-distribution is not rapid at temperature $\theta = 486^oC$, but begins already at $\theta = 200^oC$, when elastic modulus starts to reduce according to EN 1993-1-2 table 3.1 [7], and continues slowly until the failure of the whole beam at $\theta = 563^oC$. Failure temperature is about a hundred degrees higher than in linear analysis, and close to the upper limit estimate achieved with hand calculations.

6.1.4 Vulcan analysis with membrane action

Next, Vulcan model was modified so that all supports were fixed for horizontal translation. Now, axial membrane action was able to develop in beam as it loses it's stiffness and starts to act less like a beam and more like a rope, see figure 6.9.

Figure 6.9: Structure, when all supports are fixed for horizontal translation.

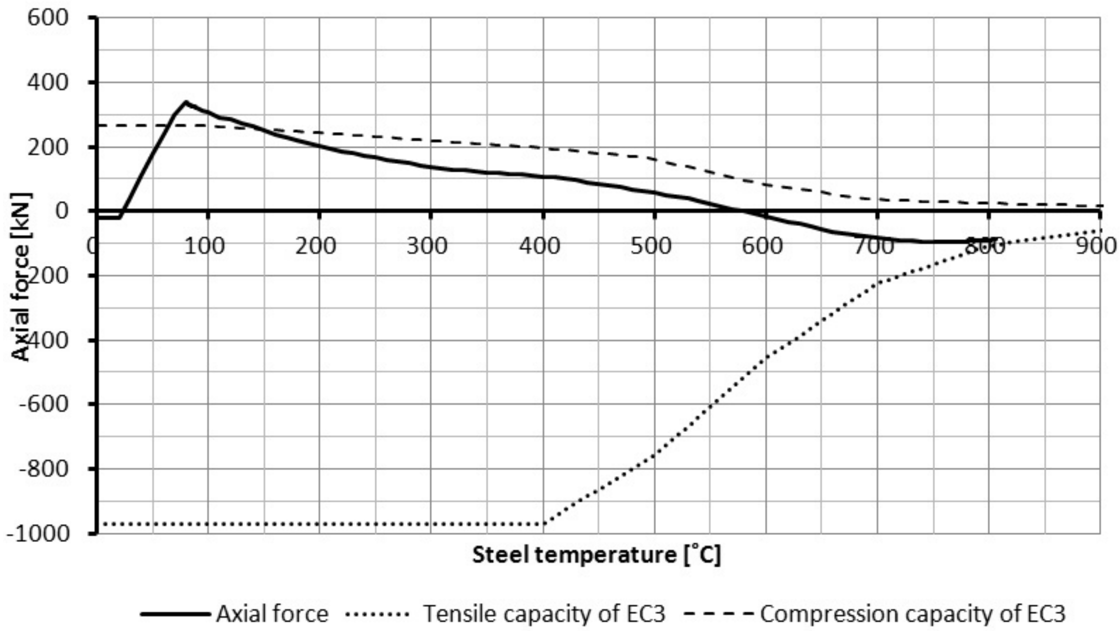

Figure 6.10: Axial force diagram.

Axial force of the beam is shown in figure 6.10. Because fixed supports prevent the thermal expansion of the beam, a significant compressive force develops in beam between temperatures 20^oC and 80^oC. At about 80^oC this force starts to release, because the beam suddenly buckles sideways, as can be seen on the Y-displacement curve in figure 6.11.

The buckling due to thermal expansion prevented Vulcan analysis to find equilirium after 80^oC, when the steel material model of Eurocode 3 was used. In reality, this buckling does not cause the beam to collapse, because axial force releases as the beam buckles, as it is caused only by thermal expansion. Using another material model supported by Vulcan, smoothed Ramberg-Osgood model [34], [35], Vulcan was able to continue analysis after the buckling and re-stabilising at 80^oC.

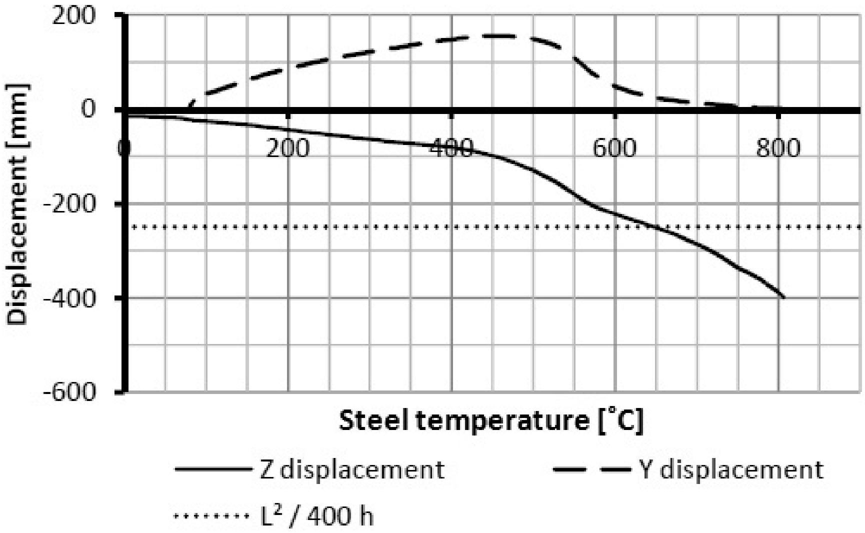

Figure 6.11: Displacement diagram.

In real world cases, this kind of buckling is not a problem, as the beam ends are not entirely fixed, but connected to columns that may bend due to the axial force in the beam. Generally when modelling isolated members or parts of structures in Vulcan, boundary conditions often cannot be modelled as they would be in linear analysis, but at least one member should be modelled between the boundary condition and the member of interest. For example when analysing the beams of one floor of a multi storey building, often the columns of one floor above and below need to be modelled, and boundary conditions applied to the column ends further away from the beams.

Figure 6.12 shows development of beam moments at spans and support as a function of time. Now beam can sustain fire until 806^oC -significantly longer than with linear or non-linear analysis, when membrane actions could not be utilised because of support conditions. In the end beam's bending capacity is almost completely lost, but beam can still sustain with it's tensile capacity -almost like a rope. At 806^oC beam's tensile capacity is finally exceeded (figure 6.10), and the beam will collapse.

It should be noted, that in order to utilise the full membrane action, not only the beam but also the beam-column connections at both ends must sustain about $350kN$ compressive force at 810^oC temperature, and $100kN$ tensile force at 800^oC temperature. The compressive capacity of typical connections is normally sufficient, but the tensile capacity at such high temperatures is likely limit the capacity of the whole structure.

The beam mid span deflection in axis Z direction is rather large, as can be seen in figure 6.11. Above a deflection limit was given in equation 2.2. This limit is also drawn in diagram of figure 6.11, and it is exceeded at temperature 650^oC. If

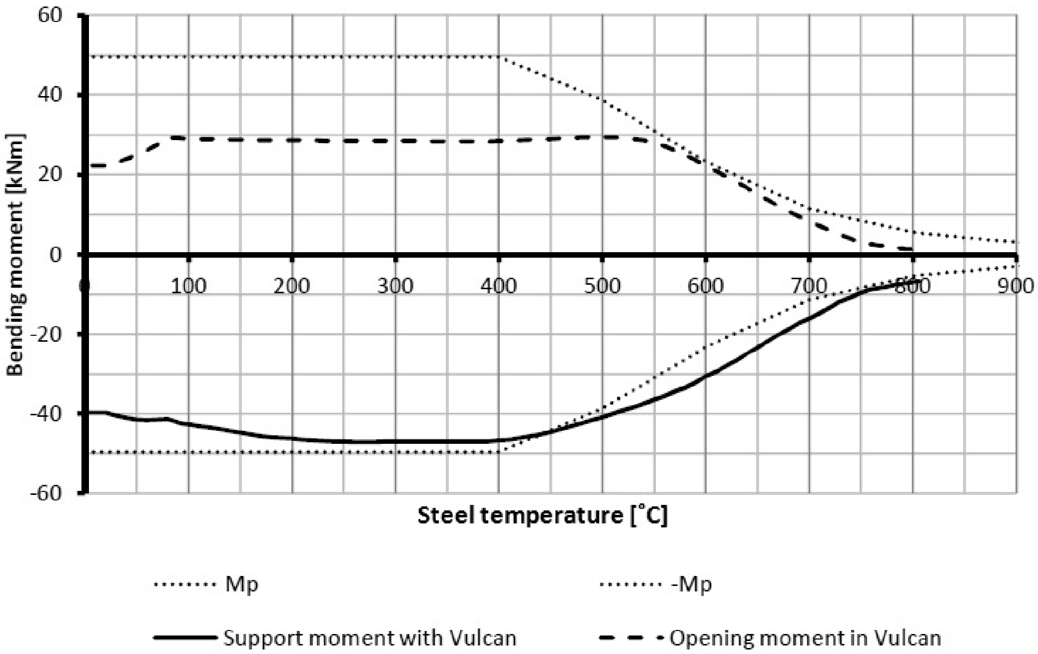

Figure 6.12: Beam moments at different temperatures: non-linear analysis using Vulcan, membrane action can form

deformations need to be limited, then the beam can sustain 650^oC temperature.

In both figures 6.8 and 6.12 support moment curve exceeds the dotted plastic bending capacity curve. Support moment starts rapidly decreasing as soon as it exceeds the limit, but theoretically it should decrease immediately as it reaches the limit moment, and never cross the dotted line, just like in figure 6.4. Reason to the inaccuracy is probably in Vulcan's solution method and can not be explained here. Results of hand calculations and linear and non-linear finite element analysis are clearly in range with each other, but because of the different methods and the simplifications in both hand calculations and finite element formulation, there is difference in results.

6.1.5 Conclusion

Failure temperatures of the three analysis methods are presented in table 6.1. It can be seen, that with the same structural model, but more accurate non-linear analysis, this structure can be validated for about a hundred degrees higher temperatures. This can sometimes be a significant result, although even 563^oC is still a rather low temperature in a fire. For example on standard fire curve the difference between gas temperatures 454^oC and 563^oC is less than three minutes.

The highest resistance of the a beam can be obtained by changing the structural model so that membrane forces can be taken into account. If the connections can take the horizontal action that forms, and deflections are not limited, the beam can sustain about 350^oC higher temperatures than validated with linear analysis. If deflections need to be limited, resistance is lower but still remarkable.

It should be noted, that this case was selected on purpose so, that the advantage of using non-linear analysis is clear. For many other structures the difference is less significant. Using linear analysis is typically on safe side, but phenomenons such as the buckling due to thermal expansion seen in figures 6.10 and 6.11 may sometimes lead it being unsafe.

Table 6.1: Failure temperatures

	Failure temperature	
	Linear / elastic	Non-linear / plastic
Hand calculations		
elastic	486^oC	
plastic		574^oC
Scia		
linear	454^oC	
Vulcan		
non-linear		563^oC
Vulcan with membrane actions		
failure limit		806^oC
$L^2/400h$-limit		650^oC

Generally non-linear analysis is more accurate, and statically non-determined structures typically have more capacity than linear analysis shows. Thermal expansion can cause significant forces in structures if it cannot expand freely. Simple hand calculations based on plastic hinges can be used to check the ultimate loading or failure temperature gained with non-linear analysis of a continuous beam, but the way how internal forces re-distribute in structure before failure is usually too hard to solve with simplified methods.

6.2 Sports hall

The second case study of this thesis demonstrates the complete work flow of fire safety design of a structure using the NFD environment. The case is an imaginary sports hall in figure 6.13, that has steel roof trusses. The goal is to find out, if the trusses can be left unprotected.

6.2.1 Building description

The hall dimensions are $25m$ x $41m$. The height of the building is about $11m$, and the bottom chord of the truss is $8m$ from floor level. The building is for sport usage, both practice and sports events, but no fairs or other multi-purpose usage.

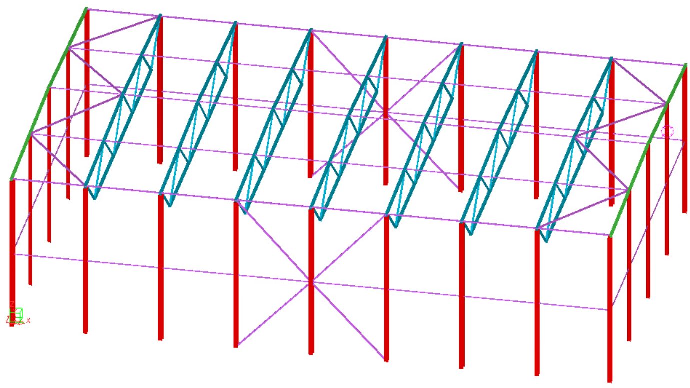

Figure 6.13: The steel skeleton of the sports hall of the case study.

The building is protected by sprinklers. According to the Finnish building code [14] the it has fire class P1. In this class, the load bearing structures must be designed for resistance period of 60 minutes, and the secondary structures such as the load bearing profiled sheets of the roof for resistance period of 15 minutes.

Without the performance based design, the standard fire curve would lead to all steel structures needing fire protection. Table 6.2 shows the needed thickness of intumescent paint if Unitherm 38091 paint had been used for fire protection according to [36].

Table 6.2: Intumescent paint thickness with Unitherm 38091

Truss chords	$1500\mu m$
Truss diagonals	$2000\mu m$
Roof ties	$2750\mu m$
Columns	$1500\mu m$
Wall ties	$1500\mu m$

6.2.2 Fire modelling

Value $347MJ/m^2$ was selected for the evenly distributed sports hall fire load from EN 1991-1-2 table E.4 [17] as the 80% fractile for fire load in "classroom of a school". Fire loads of sports usage and classroom usage were assumed to be close enough. The value is between the values for sports usage fire load in two published case studies, $196MJ/m^2$ in [20] and $600MJ/m^2$ in [22].

The values $RHR_f = 250kW/m^2$ and $t_\alpha = 300s$ were selected as the classroom values from [17] as well. With these values the RHR curve to be used as a design fire could be plotted, see figure 6.14. This curve can be used as design fire in global fire scenarios, when the fire is modelled on the whole floor area.

It has been proposed [37], that modelling the fire on the whole floor area in FDS

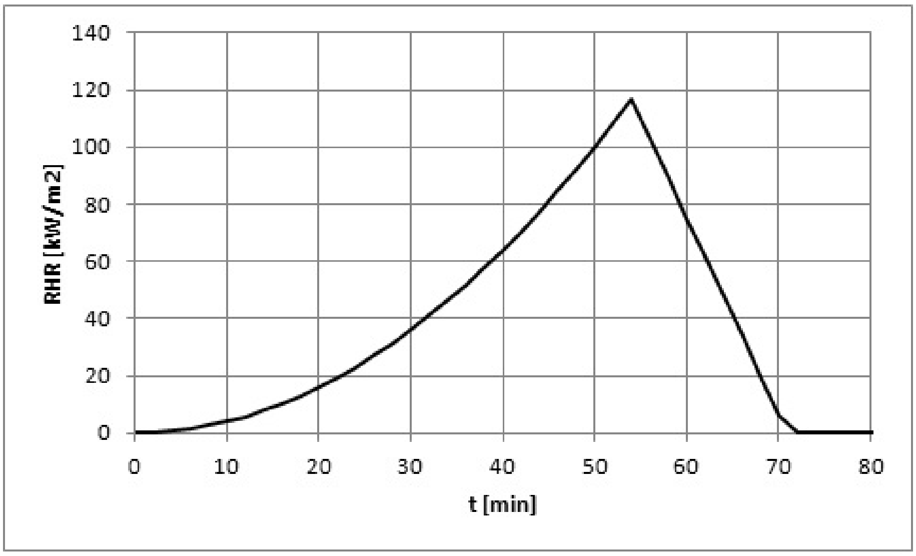

Figure 6.14: The RHR curve of the sports hall evenly distributed fire load.

is not preferable, because in real life the fire is not usually on the floor level but a bit higher, the desktop level at about $1m$ for example. Then it is closer to the structures and oxygen can freely flow below the burning surface. In FDS, the burning surface modelled like this would block the oxygen below it, so it is best to model the surface not as one surface but a "chessboard" where there is empty space between squares that burn, see figure 6.15. The fire load and RHR of these squares are then scaled in respect of the total area of the burning surface so, that the total fire load and RHR of the fire compartment does not change. [37]

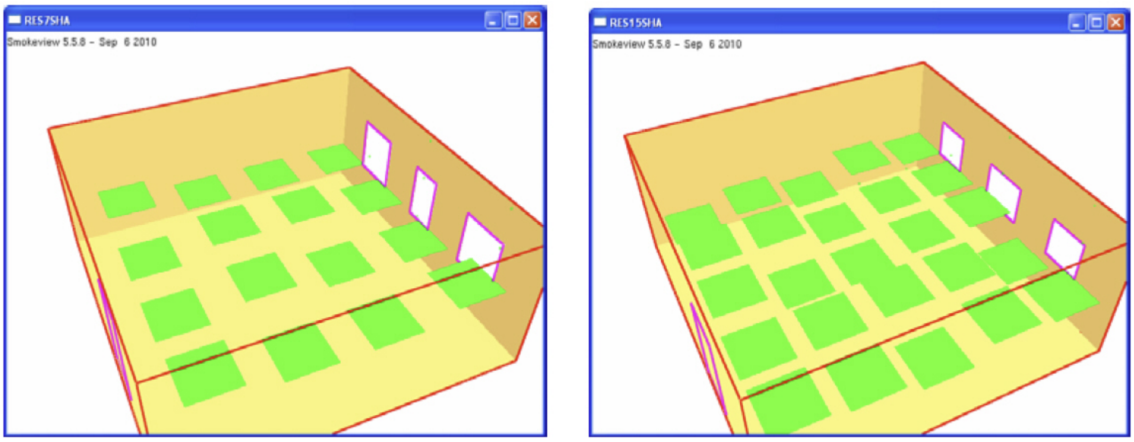

Figure 6.15: The principle of modelling the fire load of the whole compartment as a "chesboard" [37].

In addition to the evenly distributed fire load based on evaluation of the building usage, there is one possible fire load concentration that may cause a fire more severe than the design fire. The sports hall will have temporary spectator stands, that may consist of combustible materials and are high, so that the possible fire will be close to the roof trusses. The spectator stands have often been found to be critical to the structural fire safety of this type of constructions [22], [38]. The stand structure in this case was assumed to be the same as in [22], and the values presented there could be used as they were, see figure

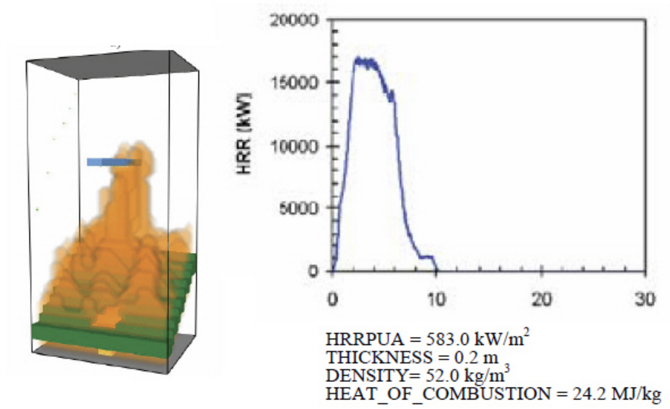

Figure 6.16: The spectator stand fire [22].

The building has two pair doors of $3,8m$ x $3,8m$ and two smaller doors of $1,2m$ x $2,4m$. The total area of windows is about $64m^2$ and they are placed near roof level and can be broken by the fire brigade for smoke exhaust. In the sensitivity analyses performed for the FDS model it was found out that in all scenarios the fire is most critical for the trusses when the windows are closed during the whole fire. Also, the model proved to be very sensitive for the amount of openings near floor level. Therefore, the amount of doors (in fire model, not real life) was doubled to four pair doors and four smaller doors to make sure the results were on safe side, and all the doors were modelled open.

6.2.3 Fire simulations

Six fire scenarios were analysed to study the resistance of steel trusses. Two types of sprinkler malfunction were taken into account: the failure of the entire sprinkler system (pump defect etc.), and the failure of the most critical sprinkler head directly above the initial fire source, allowing fire to spread to approximately $12m^2$ without sprinkler limiting the fire growth. The event tree of sprinkler malfunctions is presented in figure 6.17 together with the fire scenarios related to each branch of the event tree. Results of scenarios 2D and 2E are not presented in this study. To verify, that mid span really is the most dangerous place for local fire below the truss, also these should be analysed.

Two different scenarios of the case where sprinkler systems works properly were studied. In scenario 1A, sprinklers were not modelled in FDS, but the RHR of the fire

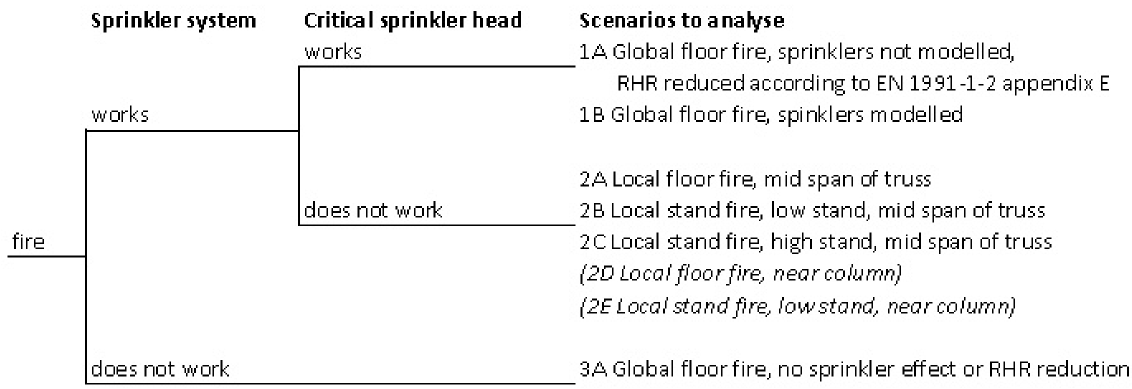

Figure 6.17: The event tree of the sports hall fire, and the analysed fire scenarios related to each branch. Results of scenarios 2D and 2E are not presented in this study, but they appear on event tree to remind that the worst case fire placement often needs to be searched by analysing more cases.

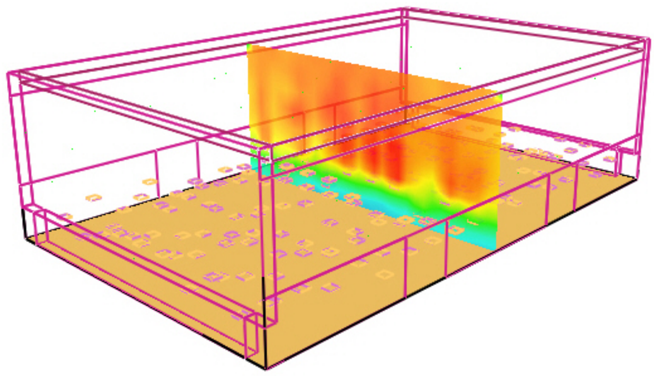

Figure 6.18: The fire simulation model of scenario 1A.

was reduced using the δ-factors of EN 1991-1-2 appendix E [17] by setting $\delta_{n1} = 0,61$. The appendix is not in use in Finland because of the National Annex, but it is the only mention in the code about taking into account the effect of sprinklers. In scenario 1B, the RHR was not reduced, but the sprinklers were modelled in FDS.

As Smokeview visualisation of the simulation of the scenario 1A is presented in figure 6.18. The temperature curves of each truss member in the scenario 1A are presented in figure 6.19, and the curves in the scenario 1B in figure 6.20. For the scenario 1A, a Vulcan visualisation showing the temperature curves of each member is presented in figure 6.18. From figures 6.19 and 6.20 it can be seen, that the difference in results between the δ_{n1} factor reduction and actually modelling the sprinklers in FDS is enormous. In order to take into account the effect of sprinklers, more understanding of the physical behaviour and it's simplified models would be required.

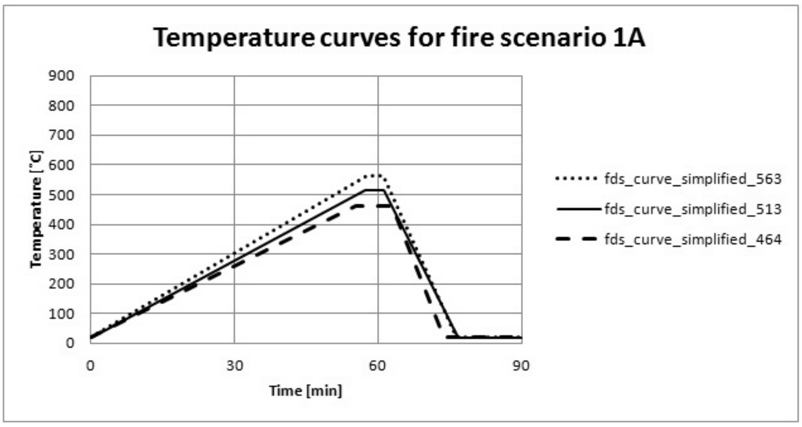

Figure 6.19: The temperature curves of the fire scenario 1A.

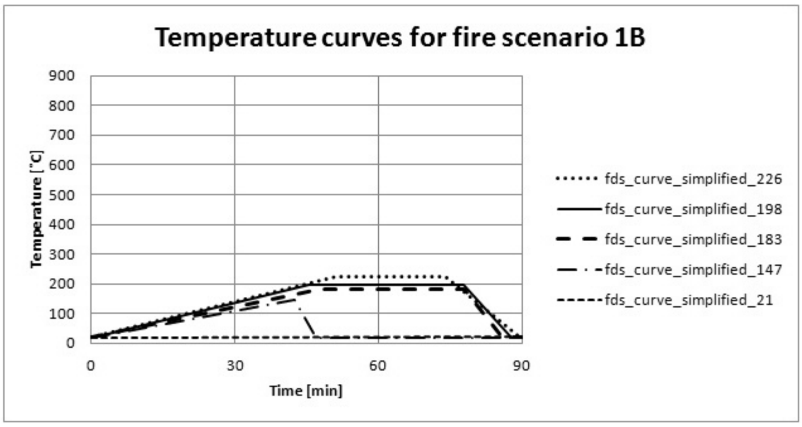

Figure 6.20: The temperature curves of the fire scenario 1B.

The fire scenarios 2A, 2B and 2C were analysed to study a case, where the most critical sprinkler head above the fire does not work, and the fire growth is not limited in a $12m^2$ area, forming an idealised local fire. In scenario 2A, the design fire is the classroom fire load on a $12m^2$ area, giving $1,4MW$ RHR at the most severe state. In scenarios 2B and 2C the design fire is the stand fire, giving a maximum RHR of about $17MW$. In 2B the stand is assumed to be $3m$ high, and in 2C $5m$ high (figure 6.21).

The gas temperatures near different roof truss members in fire scenario 2A are presented in figure 6.22. The members, to which each temperature curve is related to, are presented in figure 6.23. All of the temperatures remain very low relative to the estimated steel design temperature 600^oC. Case 2A does not require any structural analysis -the truss can sustain this fire unprotected.

Similar temperature curves for scenarios 2B and 2C are presented in figures 6.24 and 6.25. The relation between which temperature affects which member are not presented here, but they are very similar to figure 6.23, as the fire is a local fire at mid span. In scenario 2B the temperatures remain below the steel design temperature

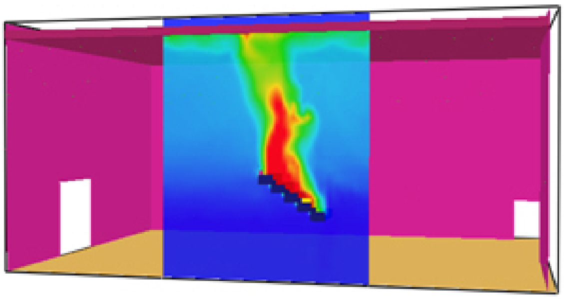

Figure 6.21: The SMV visualisation of the scenario 2C, where the $12m^2$ area of the high spectator stand is burning.

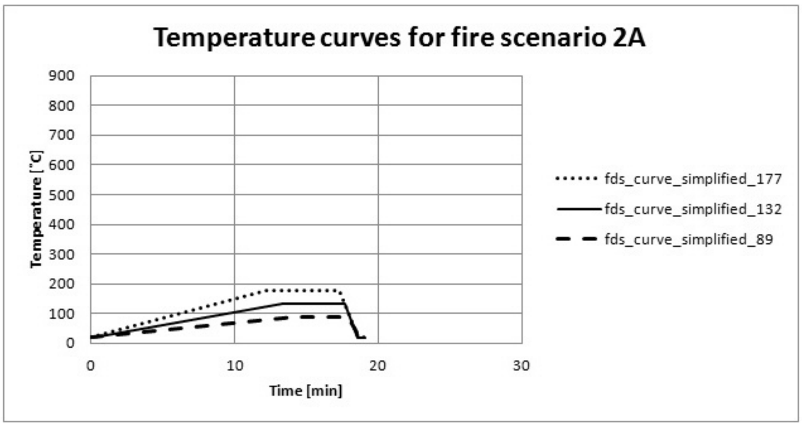

Figure 6.22: The temperature curves of the fire scenario 2A.

600^oC. In scenario 2C the 600^oC is exceeded, and the structural analysis in section 6.2.4 below show that the structure cannot sustain this fire scenario. Therefore, high spectator stands cannot be approved in the sports hall, if the roof trusses are left unprotected.

The temperature curves for scenario 3A, the global classroom fire with the whole sprinkler system not functional, are presented in figure 6.26. In this scenario, all steel members except the column bases have the same temperature curve (the one with maximum value 602^oC). The estimated design temperature for the steel is 600^oC, so a very careful structural analysis is needed to see, whether the structure can sustain the fire or not.

Often in buildings with unprotected steel structures, the whole sprinkler system malfunction leads to structural failure in fire. However, with risk assessment and probablistic analysis it is often possible to prove, that the likelihood of the whole sprinkler system failing to perform is low. In that case the risk level for the building low enough, as long as scenarios of more likely local sprinkler failure such as 2A-2C are safe.

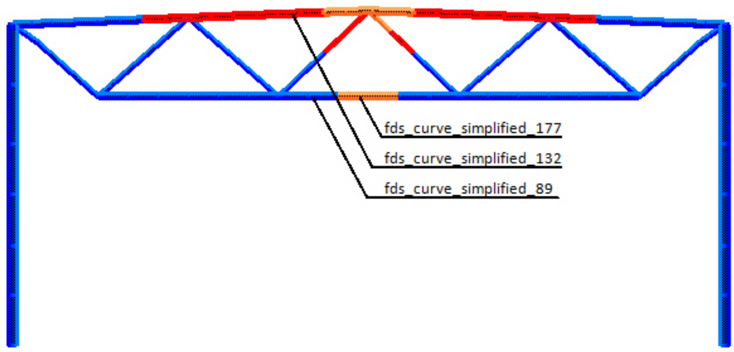

Figure 6.23: The temperature curves of each member in fire scenario 2A.

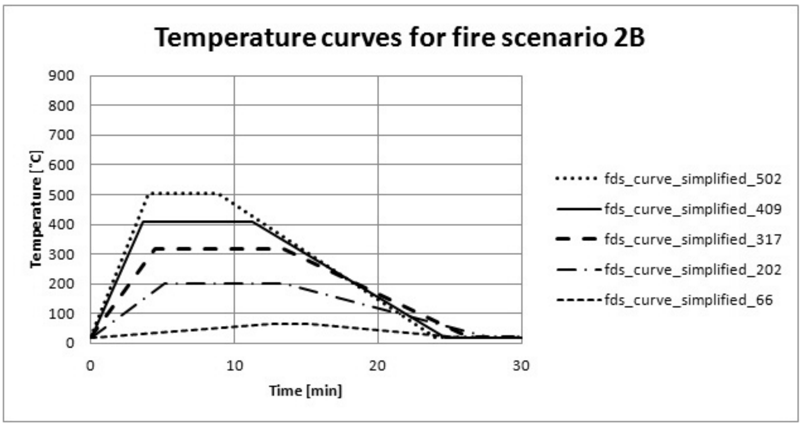

Figure 6.24: The temperature curves of the fire scenario 2B.

6.2.4 Structural analysis

Regarding all the fire scenarios presented above, 1A, 2B, 2C and 3A are interesting from the structural point of view. 1A and 3A form a pair, as the design fire is the same, but in 3A the sprinkler does not suppress the fire and the temperatures are higher. Similarly 2B and 2C form a pair, as the design fire is the same, but in 2C it is closer to the truss.

Structural analysis in Vulcan showed, that in cases 1A, 2B and 3A the structure can sustain the temperatures gained from FDS analysis. For 1A and 2B this was expected, as the gas temperatures were lower than the steel design temperature. For 3A the result is interesting, as the gas around the members was very close to the critical steel temperature, and determining whether the structure can or cannot sustain without a detailed analysis was impossible.

As Vulcan does not output any utilisation ratios of members, the only easily accessible result of the scenarios 1A, 2B and 3A for the designer is that the structure sustained. Therefore only the results from 2C are presented here in more detail.

Scenario 2C is the local spectator stand fire, where the stand is $5m$ high. The temperature curves in figure 6.25 were above the steel design temperature, so the

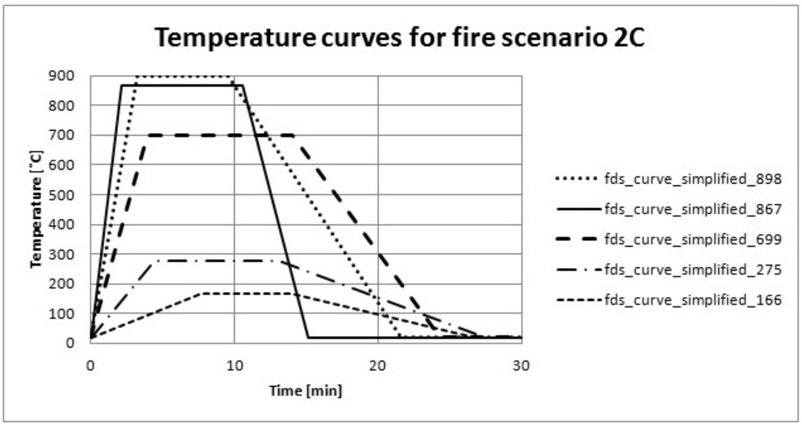

Figure 6.25: The temperature curves of the fire scenario 2C.

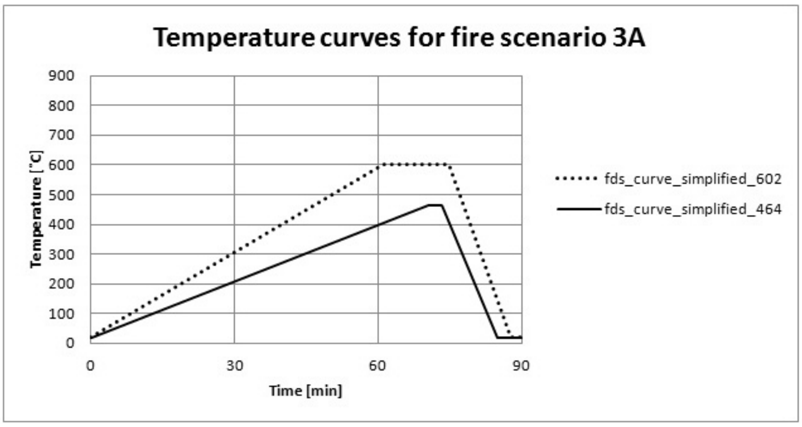

Figure 6.26: The temperature curves of the fire scenario 3A.

steel is likely to heat up too much and collapse. The Vulcan analysis showed, that the structure can sustain this fire scenario about $10,5$ minutes. Then, the truss top chord buckles near the mid node of the truss, where the chord is hottest (gas temperature reaches 898^oC). The failure of the structure is presented in figure 6.27.

6.2.5 Conclusion

The case study showed, that the roof trusses of the sports hall can be left unprotected, as long as the temporary spectator stands used in sports events are not too high (like 2C) and there are no other fire loads significantly higher than the ones examined.

From NFD-environment's point of view, it illustrated that in a real life fire engineering project several FDS simulations and structural analyses are needed, and being able to produce the needed simulation and analysis models from the building information model possible with the tools developed.

The two methods used for taking into account the effect of sprinklers gave very different results from each other. A better understanding of the effect of the sprin-

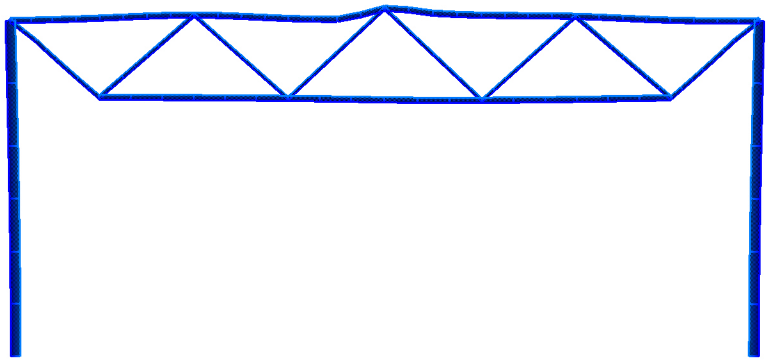

Figure 6.27: The structure in scenario 2C at 10, 5 minutes. The buckling of the top chord is visible left from the mid node. The displacements are scaled five times larger to emphasise the buckling shape.

klers would be needed to develop a practical procedure of sprinkler modelling for real projects.

7. A SIMPLE TOOL FOR PRELIMINARY STRUCTURAL FIRE ENGINEERING

The sports hall case study above showed how fire simulations may be used for determining the gas temperatures near the steel members, and how the temperature may be used in Vulcan for structural analysis. However, in performing the case study, two practical difficulties were faced:

- structural engineer normally does not have fire engineer's intuitive understanding of whether the given fire load or RHR-curve is serious or not without simulating the fire, and
- simulating the fire is very time consuming.

For preliminary design or drafting, a simpler and faster way would be preferred. There are ways to do preliminary hand calculations with the methods mentioned in section 2.3.3. However, also performing the hand calculations with pen and paper requires time.

In the example by Junnonen [18], four different simplified methods were used together in a scenario with a local fire below a roof truss. The methods were Heskestad's and Hasemi's models directly above local the fire, Alpert's model near it and 2-zone model further away (see figures 2.3 and 2.4). Assuming the worst case scenario of the fire directly below the most critical member of the truss, only Heskestad's and Hasemi's models would be needed. In the case of the flames touching the structure, the temperatures will anyway be very high and fire protection should always be applied, which is good enough result for the preliminary design. Thus, Hasemi's model is not needed for preliminary design.

As a rough estimate, the Heskestad's model alone can be used for drafting. The model is included also in the Eurocode [17]. At it's simplest form, the model can be expressed in three equations:

$$\theta(z) = 20 + 0,25{Q_c}^{2/3}(z - z_0)^{-5/3} \leq 900^o C \tag{7.1}$$

$$z_0 = -1,02D + 0,00524Q^{2/5} \tag{7.2}$$

$$Q_c = 0,8Q \tag{7.3}$$

where D is the fire diameter in meters, Q is the rate of heat release in Watts, and z is height on the axis of the flame in meters. The symbol z_0 means the co-ordinate of the natural origin of the axis of the flame, and Q_c the rate of heat that releases by convection.

In many practical cases, D only has a few typical values. In the sports hall case, all local fires had area of $A = 12m^2$, so the fire plume diameter for all local fires would be $D = \sqrt{4A/\pi} = 3,9m$. If D is constant, the temperature function can be plotted as a diagram like in figure 7.1. From this diagram a structural engineer, who knows the highest allowable steel temperature and height of the most critical steel member, can easily read the RHR level below which structure is likely to prove safe, when more detailed fire analysis is performed.

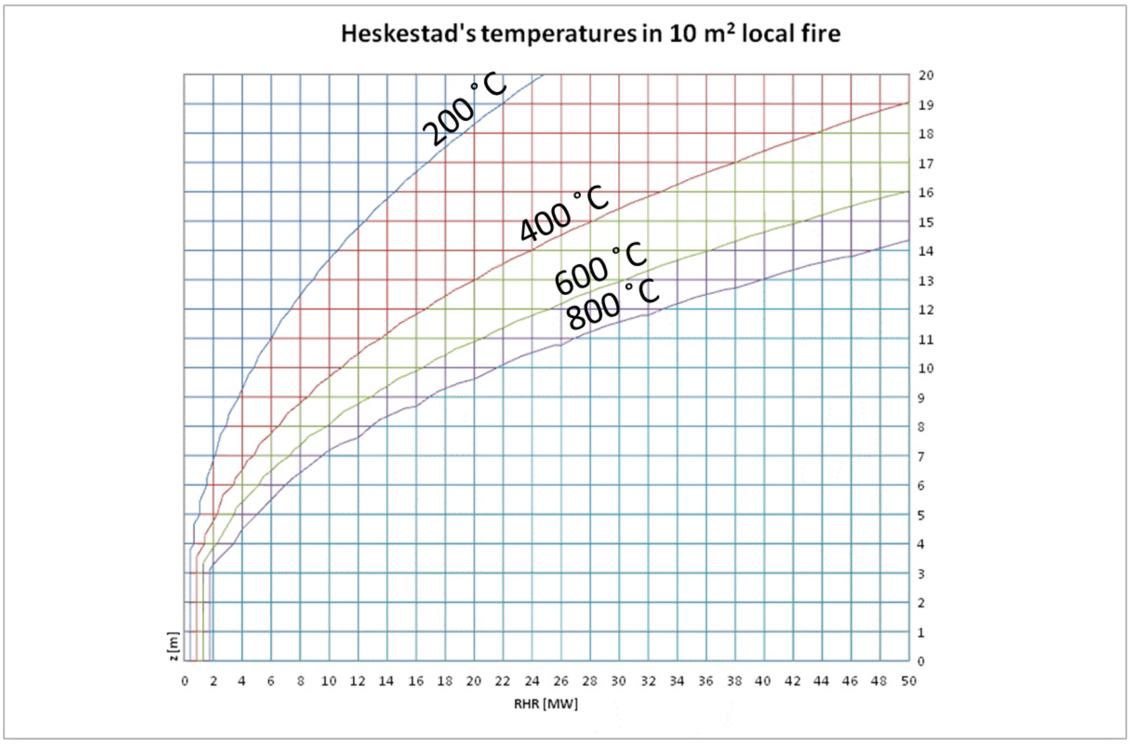

Figure 7.1: The gas temperatures at the altitude z above $10m^2$ local fires of heat release rate between 0...$50MW$

For example, if the structural engineer knows that the truss of the sports hall case study has critical steel temperatures of most members at about 600^oC, and the bottom chord is about $8m$ above the floor level, the graph can be used to determine that the structure should sustain all design fires of lower RHRs than $10MW$ (figure 7.2). A more detailed analysis (2B in section 6.2.3) showed that the structure can sustain a short $14MW$ fire near the floor level but not much higher. The initial estimate of $10MW$ was somewhat close and on the safe side.

Vice versa, the structural engineer may know have the $10m^2$ design fire of $14MW$ $8m$ below the truss, and use the graph to estimate that the gas temperature near

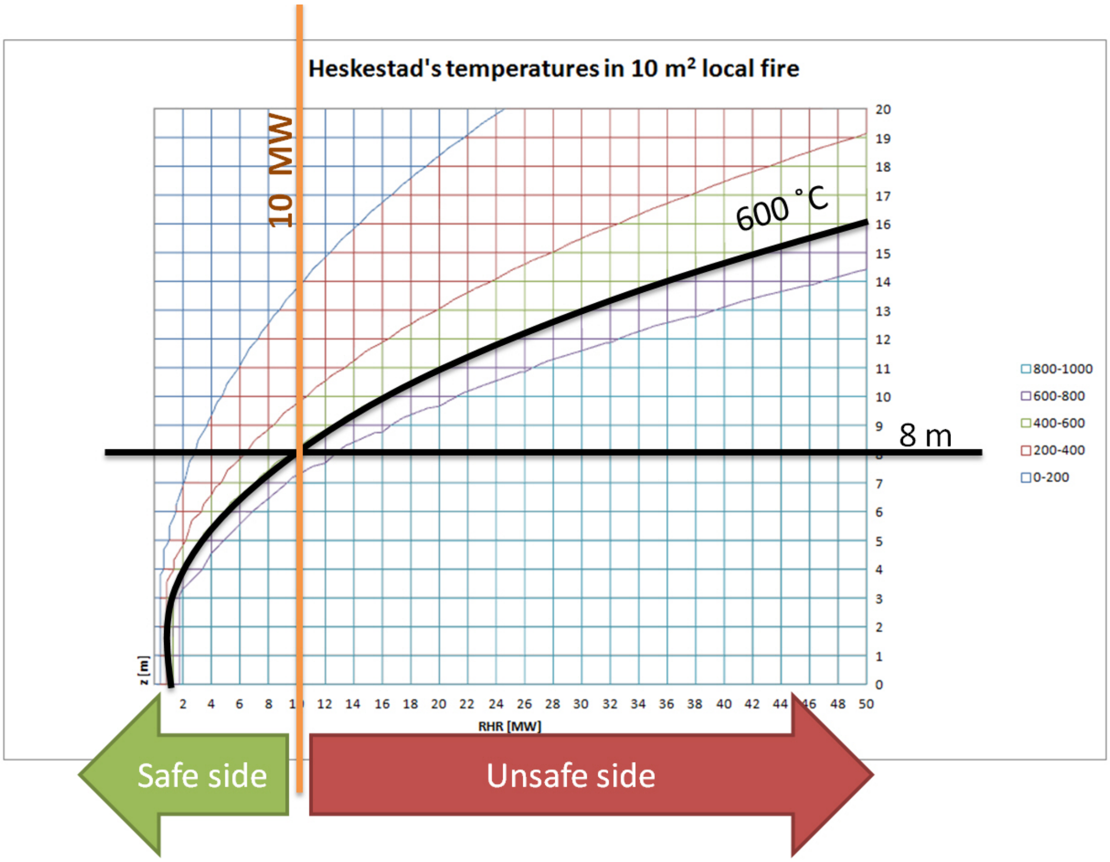

Figure 7.2: Example of estimating the safe and unsafe RHR:s for structure $8m$ above the fire, with design steel temperature of 600^oC.

truss in fire may reach 800^oC. This temperature would require fire protection for the truss. However, in not much lower temperature the truss could be left unprotected. Therefore it is reasonable to perform simulation, which in this case (2B above) would reveal that the simplified gas temperature curve does not exceed 502^oC, and fire protection is not required.

This approach is extremely simple and visual. The fire literature tends to introduce the hand calculation methods in a rather complicated equation form, and the simple graphic representation does not easily come to reader's mind. For that reason, it was presented in this study as a supporting tool that can be used for preliminary design, when the integrated fire design environment is used for the detailed design.

8. CONCLUSIONS AND FUTURE DEVELOPMENT

8.1 Conclusion

In this study, the non-linear structural analysis Vulcan was integrated with building information modelling and fire simulation. Also, a method to linearise and group the time-temperature curves derived from fire simulation was developed in order to simplify the handling of the design information and speed up the analysis.

The case studies showed, that the tools developed can be used for practical fire engineering cases. Non-linear finite element analysis is a good and accurate way of analysing statically non-determined structures in fire. Using the more accurate analysis, some structures can be validated for higher design temperatures than when using linear analysis, as shown in the continuous beam case of this study.

For statically determined structures, Vulcan does not provide any advantage over the other finite element programs capable of importing temperatures from fire simulation, such as Scia. As non-linear analysis is more time-consuming and the results are harder to examine, the linear analysis is preferable for simple structures.

8.2 Future development ideas for the field of integrating analysis of structures in fire with BIM

Integrating also the 3D component method for steel connections as a part of the system was initially set as one goal for the study, but this goal could not be reached within the project time. The previously known difficulties in understanding and visualising the Vulcan results other than displacements proved problematic in this study too, but could not be solved.

In the future development of the Tekla-Vulcan interoperability the focus should be on post-processing the Vulcan results to a more user friendly format, especially member forces and members' utilisation ratios as a function of time / steel temperature. Perhaps more importantly, a tool to automatically set the initial eccentricities relative to buckling mode to the members as the model is transferred to Vulcan, should be developed. Setting the eccentricity gives more accurate resistances for members in compression, and often simplifies the numerical solution of the non-linear FE-problem. Exaggerating the eccentricity gives results on safe side, and may help the numerics even more.

Possibility of integrating the research version of Vulcan in NFD environment was studied in addition to integrating the commercial version. It was found out, that the integration is possible, but not necessarily needed because research Vulcan is aimed for academic use only, and NFD environment has a more practical aim.

The behaviour of the connections in fire is currently very interesting topic in structural fire research. Using the 3D component method for steel connections in Vulcan would require use of the research version and possibly modification to it's source code. This is a likely research topic in the near future.

8.3 Future development ideas for the field of integrating fire simulation with BIM

Based on the experience of the case studies of this thesis, a possible future development direction for the NFD environment is a standardised and validated method of taking into account the effect of sprinklers. The two different methods tried in the sports hall case gave very different results.

Another development need derived from the case studies would be ability to control several FDS simulations from the FDS Adapter -macro of NFD environment. It currently only supports one FDS simulation per Tekla model, and the user must manually copy files between folders in order to maintain for example the six simulations that were presented in the sports hall case.

The NFD environment currently provides a good methodology for detailed or final design of structures in fire. For preliminary design or drafting, a lighter method without time consuming simulations is needed. For that, the proposed graphic presentation of the Heskestad's model presented in section 7 is a useful tool.

REFERENCES

[1] SFS-EN 1363-1. Fire resistance tests. Part 1: General requirements. CEN European Committee for Standardization, Brussels 2000.

[2] Schleich J-B., Cajot L-G., et al. Valorisation project – Natural fire safety concept. Final report. European Commission EUR 20349 EN, European Communities, 2002.

[3] Heinisuo M., Laasonen M. Product modeling, part of the fire safety concept in the future for metal structures?, Advanced Research Workshop, Fire Computer Modeling, Santander 18-20 October 2007, Universidad de Cantabria, pp. 261-272

[4] Outinen J., Heinisuo M. & Laasonen M. Combining structural design and fire simulation of steel structures with product modelling. Nordic Steel Construction Conference, September 2-4 2009, Malmö, Swedish Institute of Steel Construction, Elanders Sverige AB, Upsala. pp. 217-221. ISBN 91-7127-058-2

[5] Laasonen M., Heinisuo M., Outinen M., Lehtimäki E., Pada D. Planning environments of structures in fire. CIB 2011 Join Conference W078 / W102, Sophia Antipolis, France

[6] Hietaniemi J., Mikkola E. Design Fires for Fire Safety Engineering. VTT Working Papers 139, VTT Technical Research Centre of Finland, 2010

[7] EN1993-1-2. Eurocode 3: Design of steel structures. Part 1-2: General Rules. Strucutural fire design. European committee for standardization, 2002.

[8] Pada D. Steel Skeleton Behaviour in Decaying Fire, Licenciate Thesis, Tampere 2011. Tampere University of Technology.

[9] Leston-Jones L. The influence of semi-rigid connections on the performance of steel framed structures in fire, PhD Thesis, Sheffield 1997. University of Sheffield.

[10] EN 1993-1-8. Eurocode 3: Design of steel structures. Part 1-8: General rules. Design of joints. European committee for standardization, 2005.

[11] Heinisuo M., Laine V., Lehtimäki E. Enlargement of the component method into 3D, In Proceedings Nordic Steel Construction Conference, Malmö, Sweden, September 2-4 2009, Publication 181, LUT & SBI, pp. 430-437.

[12] Laine V. Teräsrungon liitosten jouston huomioon ottaminen integroidussa suunnittelujärjestelmässä. MSc thesis. Tampere 2008. Tampere University of Technology. (in Finnish)

[13] Lehtimäki E. Peruspulttiliitoksen jouston huomioon ottavan laskentamallin muodostaminen ohjelmallisesti liitoksen tietomallista. BSc thesis. Tampere 2009. Tampere University of Technology. (in Finnish)

[14] The National Building Code of Finland, part E1, Fire Safety of buildings, regulations and guidelines 2002. Helsinki 2002. Ministry of the Environment, Housing and Building Department.

[15] RIL 221-2003 Paloturvallisuussuunnittelu. Helsinki 2003. RIL - Finnish Association of Civil Engineers. (in Finnish)

[16] Ympäristöopas 39, Rakennusten paloturvallisuus & paloturvallisuus korjausrakentamisessa. Helsinki 2003. Ministry of the Environment, Housing and Building Department.

[17] EN 1991-1-2. Eurocode 1: Actions on structures. Part 1-2: General actions. Actions on structures exposed to fire. European committee for standardization, 2002.

[18] Junnonen J. Suojaamaton teräsputkiristikko paikallisessa palossa. MSc thesis. Helsinki 2003. Helsinki University of Technology. (in Finnish)

[19] Handbook of Fire Protection Engineering, 2nd Edition. Society of Fire Protection Engineers. National Fire Protection Association, 1995.

[20] Ympäristöministeriö, monitoimihallin paloturvallisuus. Ministry of the Environment, Housing and Building Department, Helsinki 2003.

[21] B.J. McCaffrey, NBSIR 79-1910. Purely Buoyant Diffusion Flames: Some Experimental Results. National Bureau of Standards, Washington D.C., 1979

[22] Heinisuo M., Laasonen M., Outinen J. Fire design in Europe and a case study. In Mazzollani F. (Ed.), Cost Action C26, Urban Habitat Constructions under Catastrophic Events, Final Report, CRC Press/Balkema, London, 2010, pp. 375-402. 2010.

[23] Teräsnormikortti 13/2000. Teräksen materiaalimallit mitoitettaessa palosuojaamattomia teräsrakenteita. Finnish Steel Construction Association, 2000. (in Finnish)

[24] Twilt L. Design tools for the behaviour of multi-storey steel framed buildings exposed to natural fires. 8th interim report. TNO report 2002-CVB-R06144. Netherlands Organisation for Applied Scientific Research, Center for Fire Research, Delft 2002.

[25] EN 1090 Execution of steel structures and aluminium structures. European committee for standardization.

[26] Vulcan Solutions Limited (VSL). [Online] [Cited: 6.7.2011] www.vulcan-solutions.co.uk.

[27] White Paper: Interoperability for BIM: a structural engineering viewpoint. Nemetschek Scia. 2008.

[28] Tekla Open API Developer's Guide Tekla Structures 17.0. Tekla Corporation 2011.

[29] McGrattan K., McDermott R., Hostikka S., Floyd J., 2007. Fire dynamics simulator (Version 5), User's Guide. NIST Special Publication 1019-5.

[30] Teräsrakenneyhdistys. SteelBase – Teräsrakenneprojektin oliopohjainen tiedonhallinta- ja tiedonsiirtojärjestelmä. Helsinki 1997. Finnish Steel Construction Association. (in Finnish)

[31] Plank R. E-mail discussions between 19.8.2011 - 3.10.2011.

[32] Lehtimäki E. STSM Scientific Report. COST action TU0904 Integrated Fire Engineering and Response. Non-published report 2012.

[33] ENV 1993-1-2. Eurocode 3. Design of steel structures. Part 1-2. General rules. Structural fire design. European committee for standardization 1995.

[34] Ramberg W, Osgood W. Description of stress-strain curves by three parameters. National Advisory Committee for Aeronautics 1942. Technical Note 902.

[35] Shepherd P. The performance in fire of restrained columns in steel framed construction. PhD thesis. Sheffield 1999. University of Sheffield.

[36] Teräsrakenneyhdistys, Varmennettu käyttöseloste TRY-104-2009, Sika Unitherm 38091 -palonsuojamaali, Finnis Steel Construction Association, 2009.

[37] Laasonen M., Heinisuo M., Outinen J., Hietaniemi J. Systematisation of fire loads in Ruukki´s integrated fire design. Tampere University of Technology, 2010. Internal research notes.

[38] Pada D. Simulation and Study of Natural Fire in a Wide-Framed Multipurpose Hall with Steel Roof Trusses. Acta Polytechnica Vol. 49 No. 1/2009